ICEPIC

In-Circuit Emulator

User's Guide

Microsoft Windows Version

RF Solutions Ltd, & NeoSoft, Inc.

Contents

Contents 2

Chapter 1: Introduction 6

Welcome 6

Product Definition 7

ICEPIC Software 9
Internet Access 9

Host Computer Requirements 9

Warranty Registration 10

CE Compliance 10

Chapter 2: Installation 11

Hardware Installation 11

Removal of Daughter Board; 11
Installation of Daughter Board; 11
Configuration of Daughter Board; 11

Software Installation 12

Chapter 3: Menu Commands 13

Quick Reference 13

File Menu (ALT+F) 16

Load Object File ... (ALT + F L) 16
Save File Registers ... (ALT + F R) 17
Save To Intel Hex File ... (ALT + F I) 17
New Source (ALT + F N) 17
Open Source ... (ALT + F O) 18
Close (ALT + F C) 18
Save (ALT + F S) 18
Save As ... (ALT + F A) 18
Print (ALT + F P) 19
Print Setup ... (ALT + F R) 19
Exit 19

Edit Menu (ALT + E) 20

Undo (ALT + E U) 20
Cut (ALT + E T) 20
Copy (ALT + E C) 20
Paste (ALT + E P) 21
Find ... (ALT + E F) 21
Find Next (ALT + E N) 21
Replace (ALT + E R) 22
Match Bracket (ALT + E B) 22
Edit Tab Stops (ALT + E S) 22

Configure Menu (ALT + C) 23
Emulator System ... 23
Serial Port ... 24
Load Workspace ... 25
Save Workspace ... 25

Debug Menu (ALT + D) 26
Go (ALT + D G) 26
Halt (ALT + D H) 27
Reset (ALT + D R) 27
Step Into (ALT + D S) 27
Step Over (ALT + D O) 28
Step Out (ALT + D U) 28
Run To Cursor (ALT + D N) 28
Animate (ALT + D A) 28
Toggle Breakpoint (ALT + D T) 29
Break Points ... (ALT + D B) 29
Clear All Breaks (ALT + D C) 30
Add Watch Variable (ALT + D W) 30
Add Quick Watch (ALT + D Q) 32
Modify Register ... (ALT + D M) 33
Modify Code Memory ... (ALT + D F) 33
Reset System (ALT + D Y) 33

Qualifiers Menu (ALT + Q) 34
Toggle Break Point (ALT + T B) 34
Break Range (ALT + Q B) 34
Toggle Trace Point (ALT + Q A) 34
Trace Range (ALT + Q T) 35
Toggle Trigger Out (ALT + Q G) 35
Trigger Out Range (ALT + Q O) 35
Clear All Qualifiers (ALT + Q C) 35

Tools Menu (ALT + T) 35
Assemble (ALT + T A) 36
Compile (ALT + T C) 36
Programmer (ALT + T P) 36
Editor (ALT + T E) 36
Calculator (ALT + T L) 36

Options Menu (ALT + O) 37
Project... (ALT + O P) 37
Debug... (ALT + O D) 38
Code Download ... (ALT + O L) 40
Real Time Trace... (ALT + O R) 42
Directories... (ALT + O I) 43
Colour... (ALT + O C) 43
Font ... (ALT + O F) 43
Tools ... (ALT + O T) 43

View Menu (ALT + V) 45
Toolbar (ALT + V L) 45
Status Bar (ALT + V B) 46

Code Memory (ALT + V C) 46
SFR (ALT + V R) 47
Data File Registers (ALT + V D) 48
Call Stack (ALT + V S) 48
EEPROM Data (ALT + V E) 49
Real Time Trace (ALT + V T) 49
Watch Window (ALT + V W) 50
Debug Window (ALT + V G) 50

Window Menu (ALT + W) 51
Cascade (ALT + W C) 51
Tile Horizontally (ALT + W H) 51
Tile Vertically (ALT + W V) 51
Arrange Icons 51

Help Menu (ALT + H) 52
Index (ALT + H I) 52
Using Help (ALT + H U) 52
About ICEPIC Emulator (ALT + H A) 52

Chapter 4: Diagnostics 53

Appendix A : Text Editor & Source Level Debug 56

Source Debug Features 56

Text Searching 58

Editor Keyboard Commands 62

Appendix B : Programming ICEPIC 63

Introduction 63

Exposed Objects 63
ProcessorCmd 64
GetPC 65
GetRam 66
SetRam 67
IsHalted 68
DownLoad 69
FillCodeMem 70
BreakPoint 71
ClearBreaks 72
GetEEprom 73
SetEEprom 74

Excel Automation Example 75

Visual Basic Code (VBA) 75

Automatic Data Links 77

Appendix C: Additional Features of ICEPIC2 79

Appendix D: Troubleshooting 81

Appendix E: Serial Communication 82
Appendix F: Data Types and Formats 84
Appendix G : Power Supply 85
Warning! 85
Appendix H : User Configuration Links 86
OSC Mod Config 86
MCLR 86
OSC Input - Internal or External Source 86
ICEPIC & ICEPIC2 Daughter Board Link Configuration 87
ICEPIC-Jnr-5X Link Configuration 88
ICEPIC-Jnr-16 EmulatorBoard Link Configuration 88
Appendix I : Emulation Limitations 89
Warning! 89
Processor Specific Limitations 89
ICEPIC System Specific Limitations 89
ICEPIC Oscillator/Clock Limitations 90

RF Solutions Ltd.
Pannetts Building , Railway Lane,
Lewes, East Sussex. BN7 2AQ
England
Tel : +44 1273 488 880
Fax : +44 1273 480 661
Email icepic@pavilion.co.uk
http://www.rfsolutions.co.uk

Chapter 1: Introduction

Welcome

Thank you for purchasing the ICEPIC In Circuit Emulator system.

The ICEPIC range of In Circuit Emulators, are fully modular, Real Time In-Circuit Emulator (ICE) systems, for use with Microchip Technology's PIC series of Microcontrollers.

The original ICEPIC was a concept from two Ex-Microchip personnel. Amar Palacherla, (formerly Development kit designer and original designer of Microchip's PIC-Master In Circuit Emulator) and John Fairall. (formerly Applications Manager and Sales Manager for ASIA Pacific). Our target was to produce a fully functional, In Circuit emulator system with an easy operating IDE (Integrated Development Environment). The aim was to market the tool at an affordable price, particularly when compared to the cost of the PIC microcontroller itself.
Developed under contract from Microchip, the ICEPIC was designed as a modular system, to enable future upgrades as new PIC microcontrollers are released.

There are four ICEPIC systems to choose from, ICEPIC-Jnr-5X, ICEPIC-Jnr-16, ICEPIC and ICEPIC2. All ICEPIC systems have been designed to be compatible with Microchip's own software tools, and will continue to be in the future. New daughterboards are in design and we look forward to providing emulation for new PIC microcontrolllers as they are released..

The salient features of the emulator are :

- Real Time Non-Intrusive Execution
- 8K words of emulation Memory (expandable to 16K)
- Full speed real time emulation
- Single step
- Automated single steps
- Source level debugging (both assembly & C)
- Symbolic debug capability
- Same user interface for all PIC16CXX processors
- 8K Hardware break points
- Custom watch points
- Watch and modify multi-byte registers
- Microsoft Windows 3.1 Compatible
- OLE Automation for custom programming

In addition ICEPIC2 offers:

- 4K words of Real Time Trace Memory
- Hardware Trigger Output
- PIC17Cxx Emulation

We trust that you will find it to be an invaluable tool and welcome any comments you may have.

Product Definition

ICEPIC-Jnr-5X

The components of the system are:

- Emulator Circuit Board.
- RS-232 Cable.
- Probe Header Cables.
- ICEPIC software.

Emulator Circuit Board

The emulator pod carries all the circuitry of the emulator. It consists a single board. Various optional accessories are available, such as an enclosure for the system and power supply.

The circuit board contains various jumper links. For a description of the function of these please refer to Appendix “ICEPIC-Jnr-5X Board Configuration”, at the back of this manual.

ICEPIC-Jnr-16

The components of the system are:

- Emulator Circuit Board.
- Low Cost Personality Board
- RS-232 Cable.
- ICEPIC software.

Emulator Circuit Board

The emulator circuit board carries all the circuitry of the core emulator. It contains the core emulation chip (16C02) and the logic circuitry. The combination of the core and peripheral emulation chipset provide the complete emulation of the target processor and its peripherals.

Personality Board

The personality board plugs on to the main emulator board. This board contains the peripheral emulator chip, thus providing the peripheral functions for the system to enable emulation.

This board provides connection to the header cable which connects to the target application board.

The personality board contains various jumper links. For a description of the function of these please refer to Appendix “ICEPIC-Jnr-16 Board Configuration”, at the back of this manual.

ICEPIC, and ICEPIC2

The components of the system are:

- Emulator Pod.
- Personality Daughter Board
- RS-232 Cable.
- ICEPIC software.

Emulator Pod

The emulator pod carries all the circuitry of the emulator. It consists of two boards. The main emulator board containing all the logic circuitry, and the personality daughter board.

Personality Daughter Board

Within the Emulator Pod, the personality daughter board plugs on to the main emulator board. This board contains the core emulator processor (e.g. 16C01/2) and the appropriate emulation device PIC16CXX (e.g. 16C71-ME), thus providing the peripheral functions for the system to enable emulation. Note the 14pin XTAL module, may be replaced with an equivalent type at any specific frequency the user desires (frequency range limits 32khz - 20MHz)

This board provides connection to the header cable which connects to the target application board.

The daughter board contains various jumper links. For a description of the function of these please refer to Appendix “Daughter Board Configuration”, at the back of this manual.

ICEPIC Software

The ICEPIC Software runs under Microsoft Windows 3.11 or higher environment and provides a user friendly interface.

A Windows95 is under development at the time of this manual being written. This will provide a 32bit interface operating system for use under Win95 and Win NT.

Internet Access

The latest version of ICEPIC Windows software is available for download from the internet WEB page;

http://www.rfsolutions.co.uk

Host Computer Requirements

We recommend the following minimum requirements;

- 386 or greater PC Compatible.
- VGA or higher resolution (recommend 1024 x 764 SVGA)
- 4Mbytes of RAM (recommended 8Mb)
- Microsoft Windows 3.1 or higher (recommend Windows For WorkGroups)
- DOS 3.3 or greater (recommend DOS 6.22)
- At least 4Mbytes of Free Hard Disk Space
- 1 Free Serial Port - See Appendix “Serial Communication”.
- A pointing device. e.g. Mouse

Warranty Registration

The supplier does not warrant or guarantee that the operation of this product will be uninterrupted or totally error free.

To the extent permitted by law;

1. The supplier hereby excludes all liability for any loss or damage, whether direct or indirect, sustained as a result of operating this product (save to the extent that liability shall not be excluded for any personal injury or death arising as a result of the negligence of the supplier or the authorized representative thereof).
2. Excludes all liability for any loss or damage (other than personal or death) arising out of contract or tort or in any other manner and whether or not arising out of negligence of the supplier or the authorized representative thereof.

No information provided by the supplier and/or contained within this product regarding device applications or otherwise is intended to be relied upon in any manner by persons or bodies using this product and all such information is subject to being altered or updated at any time. No representation or warranty is given and no liability is assumed by the supplier with regard to the accuracy or use of such information, or with regard to the infringement of patents, copyright or other intellectual property rights arising out of the use of such information.

This product is warranted against defects and workmanship for a period of 90 days from date of shipment. During the warranty period, the supplier will, at his discretion, repair or replace the product if it proves defective either on receipt or on normal usage during the said period.

This product must be returned to the supplier for warranty service within the stated period. Warranty is only valid after receipt by the manufacturer of a completed registration card. The buyer shall pay all shipping costs and other charges or assessments for the product by the supplier.

Please fill out and return the warranty card.

By doing so you will validate the product warranty, receive any new product updates and notification of interim releases that may become available.

CE Compliance

EPROM Emulators and similar test equipment (e.g. ATE), are often connected to exposed or unscreened circuit boards which themselves can cause EMC emissions likely to affect other sensitive apparatus. Extreme care must be observed when handling and operating the equipment . Static handling procedures must be observed at all times.

Emissions may vary depending on the configuration of the equipment.

Emissions are less when the emulator is configured without the probe header cable connected.

An increase in emissions is likely when the emulator probe header cable is connected.

It is the user’s responsibility to be aware of the inherent danger of the effect of emissions on other equipment within range of these EMC emissions. It is the user’s responsibility to take corrective action when operating apparatus likely to be sensitive to electromagnetic emissions (e.g. radio equipment) within the vicinity of the emulator.

Chapter 2: Installation

Hardware Installation

Listed below is the procedure for the correct power-up sequence,

1. Ensure the target cables are properly connected and the Oscillator Module is connected. Note the OSC Module is inserted with the correct orientation!
2. Connect the ribbon cable header into the target application. Note: Pin 1 of the ribbon cable is coloured “red”, Pin 1 of the DIP Header connector is adjacent to the “red” side of the cable.
3. Connect the ICEPIC-In circuit emulator Power supply.
4. Connect the RS-232 cable between host computer and the rear panel of the ICEPIC-In circuit emulator pod.
5. Turn on the power to the host computer
6. Turn on the power to the ICEPIC-In circuit emulator pod.
7. Turn on the power to the target application.
8. It is recommended that the MCLR pin on the target board is not held low.

Removal of Daughter Board;

1. Ensure all power is disconnected.
2. Disconnect the RS-232 cable to the host computer and header cable to the target application.
3. Place the ICEPIC-In circuit emulator pod upside-down. Unscrew the two screws to release the bottom half of the plastic case assembly and front panel.
4. Turn the pod upright. Carefully ease off the daughter board (the top PCB) from the two “DIN41612” type connectors.

Installation of Daughter Board;

1. Complete the removal of the existing daughter board.
2. Installation of a daughter board is a simple push fit onto the main board, connecting only by the two “DIN41612” type connectors. Ensure the correct polarity of the daughter board.
3. Ensure the Emulator XTAL and Target header cable are securely inserted.

Tip: Before replacing the plastic box, you may wish to configure Jumper Link3 (MCLR).

1. Slide in the front panel into the case slots.
2. Replace the bottom half of the plastic case assembly, ensure the target cable(s) pass through the slot in the front panel. Insert the two case screws.

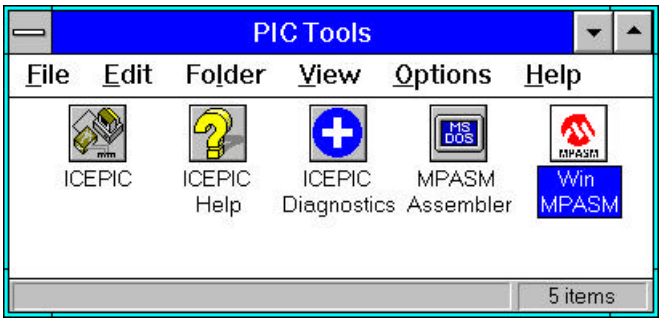
Configuration of Daughter Board;

There are several Jumper Links which are used to configure your daughter board. These are all preset during manufacture.

Please refer to Appendix “User Configuration Links” for details on these jumper links.

Software Installation

1. Insert the setup disk (disk #1) into a floppy drive
2. Select the Windows Program Manager, File Manager, or other shell such as the ones in Norton Desktop for Windows or PC Tools for Windows.
3. Choose Run from the File menu.
4. Type "A:\SETUP" or "B:\SETUP" in the box entitled Command Line, then choose OK.
5. Follow the on-screen prompts to install the software into a specified directory of your hard disk.
6. When prompted to enter the destination directory, type in the desired directory of your hard disk (e.g "C:\ICEPIC"). If the directory does not exist, it is created automatically by the program.
7. After copying all the desired files, the program checks the number of serial ports available on your computer and prompts you to select the desired COM port to which the ICEPIC-In circuit emulator is connected.
8. When prompted to insert Disk # 2, remove Disk # 1, insert Disk # 2 and press OK
9. A new program group "PIC16CXX Tools" is created as shown below



10. The software installation is completed
11. Double Click on the "ICEPIC" Icon to start ICEPIC. Make sure the ICEPIC hardware is connected to the selected COM Port and check the power

Chapter 3: Menu Commands

Quick Reference

Each menu command is briefly described in the table below. A detailed description of each command is given later in this chapter

File Menu	
Load Object File ...	Downloads the selected object code to emulator
Save File Registers ...	Saves all the File Registers of the processor to the selected file
Save To Intel Hex File ...	Saves the code memory to an Intel Hex File
New Source	Opens an empty source code document
Open Source ...	Opens an existing source code document
Close	Closes the current source document
Save	Saves the current document
Save As ...	Saves the document to a new file
Print ...	Print the current source document
Print Setup ...	Sets the printer and its options
Exit	Quits the program and holds the emulator system in reset state

Edit Menu	
Undo	Undo the last action in the editor
Cut	Cut the current selection
Copy	Copy the selection to clipboard
Paste	Paste the contents of clipboard
Find	Search for text
Find Next	Repeat the last search
Replace	Replace a word by another word
Match Bracket	Shows the matching bracket
Tab Stops	Sets tab length in the editor

Configure Menu	
Emulator System ...	Selects the target processor and various settings of the processor
Serial Port ...	Selects the desired COM port and baud rate
Load Workspace ...	Loads a previously saved workspace setting from the selected file
Save Workspace ...	Saves the current workspace setting to a file

Debug Menu	
Go	Run the processor in real time
Halt	Halt the processor
Reset	Reset the processor

Debug Menu	
Step Into	Single step the processor
Step Over	Step over a subroutine or function call
Step Out	Exit out of the current function call
Run To Cursor	Run upto cursor in editor or selection in code window
Animate ...	Run the processor in continuous single step mode
Toggle Breakpoint	Add or remove breakpoint at current location
Breakpoints ...	Open a dialog box to perform various breakpoint functions
Clear All Breaks	Remove all breakpoints
Add Watch Variable ...	Opens a dialog box to setup watch points
Add Quick Watch	Adds a watch point using the word under cursor
Modify Register ...	Opens a dialog box to modify a file register
Modify Code Memory ...	Opens a dialog box to modify code memory
System Reset	Reset the system, placing the emulator system in a power-up state

Tools Menu	
Assemble	Run the assembler
Compile	Run the compiler
Programmer	Run the programmer
External Editor	Run the external editor program
Calculator	Run the calculator program

Options Menu	
Project ...	Set various project options
Debug ...	Set various debug options of the target processor and emulator functions
Code Download ...	Sets various actions to perform on code download
Real Time Trace ...	Set real time trace options
Directories ...	Set the path of various directories
Color ...	Set text and background colours
Font ...	Set *font* sizes
Tools ...	Configure program names for the tools menu

View Menu	
Tool Bar	Enables or disables the main tool bar
Status Bar	Enables or disables the status bar
Code Memory	Display the code memory child window
SFR	Display Special Function Registers
Data File Registers	Display all file registers of the processor
Call Stack	Display the processor's stack
EEPROM Data	Display the processor's EEPROM data
Real Time Trace	Display the real time trace data

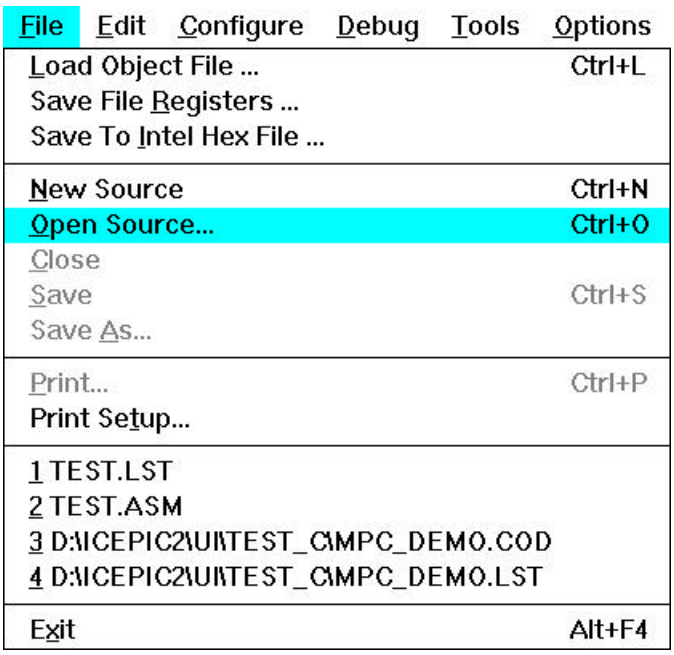
View Menu	
Watch Window	Display the watch window
Message Window	Displays errors/warnings generated from other tools (MPASM, mpc, etc)
Debug Window	Enable and display the debug window

Window Menu	
Cascade	Arrange all child windows by cascading them
Tile Horizontally	Tiles all child windows horizontally
Tile Vertically	Tiles all child windows vertically
Arrange Icons	Arrange all the icons

Help Menu	
Index	Opens the on-line help with index in view
Context Help	Places the ICEPIC under context help mode.
Using Help	Shows how to use Windows help system
Internet & World Wide Web	Opens ICEPIC help system, showing Internet Information
Programming ICEPIC	Opens ICEPIC help system and shows how to program ICEPIC
About ICEPIC ...	Displays information about your emulator system

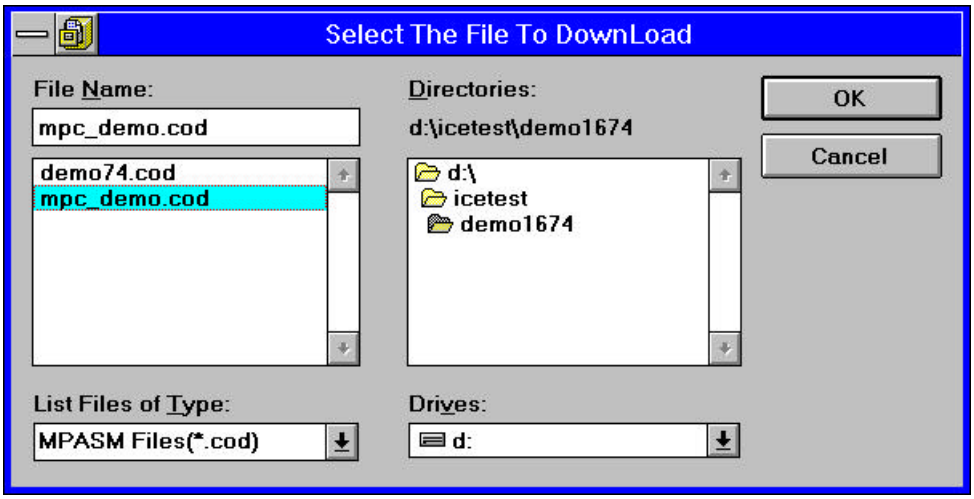
File Menu (ALT+F)

The File Menu has the following commands which are explained below



Load Object File ... (ALT + F L)

Loads Object files generated by MPASM/MPC and downloads code memory to emulator. The default file type is "*.COD". Optionally one may choose Intel Hex (*.HEX) files generated by either MPASM or another tool.



The COD file is automatically generated by the MPASM assembler and MPC "C" compiler. The download to emulation memory is aborted if serial errors occur or if the COD file is invalid. After successful download to the emulator system, the CODE Window is updated with new data.

Upon download, certain actions to be performed (like opening listing file for source level debug) can be set using the "Options Code Download..." menu.

NOTE *:*

Symbolic & Source Level debug functions are available only when using COD files generated by MPASM/MPC

Accelerators:

Use Toll Bar Button

CTRL + L

Save File Registers ... (ALT + F R)

Saves all Data File Registers (data memory window) in a user specified file. The data is saved in a text file (ASCII format) and can be edited using any standard editor. The data is saved as HEX values.

This feature is useful for documentation and debug. The default extension of the file is RAM.

A typical output file is shown below :

Data File Registers (in Hex) Of PIC16C57

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	B3	A6	DB	3C	87	0C	3E	99	24	5E	0D	1C	06	B7	47	DE
30	5D	D4	CB	FC	96	F5	45	3B	13	0D	89	0A	1C	DB	AE	32
40	29	23	BE	84	E1	6C	D6	AE	52	90	49	F1	F1	BB	E9	EB
50	AD	4F	14	F2	44	40	66	D0	6B	C4	30	B7	32	3B	A1	22
70	2C	80	7E	C5	99	D5	E9	80	B2	EA	C9	CC	53	BF	67	D6

Save To Intel Hex File ... (ALT + F I)

Saves code memory to a file in Intel Hex File Format. This feature is useful to download code to programmers. Although Intel Hex File is generated by the assembler, sometimes while debugging, one may make a few quick changes in the emulator system, and save these changes to an Intel Hex File for testing and experimental purposes.

The default extension of the file is HEX.

New Source (ALT + F N)

Opens a new source (text) file to edit using the built in Internal Editor. The file can be edited and saved just like a normal text file. This is primarily intended to edit source code files, save them, compile or assemble and then download for debug. This feature is part of the ICEPIC's Integrated Development Environment (IDE).

After a successful compile/assemble of this file, this same editor window is used as a source level debug window.

Accelerators:

CTRL + N

Open Source ... (ALT + F O)

A standard File Open Dialog Box appears. Select the desired text file to open the file for editing or Source Level Debugging using the internal editor. Any text file can be opened. The default file extensions are listed in File Types. However any extension can be forced open.

For further information see chapter on built in editor.

Accelerators:

CTRL + O

Close (ALT + F C)

Closes the current source code/text window. If the file is modified, the user is prompted for a save option. Press "Cancel" button to abort this action. Press OK to continue to close down after saving. If the file is a newly created file ("untitled"), then a dialog box (standard file save as dialog) appears asking for a new file name.

Save (ALT + F S)

Saves the file in the current window using the existing file name. The previous contents of the file are overwritten. To save the file to a new location, use Save As menu command.

Accelerators:

CTRL + S

Save As ... (ALT + F A)

Opens a “File Save As” dialog box to save the contents of the currently active editor window to the desired file.

Print (ALT + F P)

Prints the contents of the currently active editor window using the active printer. The whole document is directed to the printer. To change the default printer or its settings, use "Print Setup ..." menu command.

Hot Key :

- **Use the toolbar button**

Print Setup ... (ALT + F R)

Use this command to modify the default settings of the currently selected printer or to select a new printer.

Exit

Ends the emulation session. The current settings of the workspace (window positions, colours, etc) are saved to the default INI file and are restored automatically later when a new session is begun.

One may also exit by either of the following :

- Press ALT + Spacebar and choose close menu.
- Double click on the program's Control menu.
- Use the Hot Key ALT + F4

NOTE: When the program is quit, the emulator system, if connected, is held under reset condition.

Edit Menu (ALT + E)

This menu has the following commands which are fully described below.

Undo (ALT + E U)

Undo the last action in the text editor window. This function is used to return to the previous state when an undesired action has been performed.

This function is available only in the text editor window(s).

Hot Key :

- **CTRL + Z**

Cut (ALT + E T)

Cuts the selected text in the text editor window and places it on the Windows Clipboard. This text is placed on the clipboard and can be pasted into other text editor window or into other applications.

This function is available only in the text editor window(s).

Hot Key :

- **CTRL + X**

Copy (ALT + E C)

Text Editor Window

Copies the selected text in the text editor window and places it on the Windows Clipboard. This text is placed on the clipboard and can be pasted into other text editor window or into other applications.

Data File Register Window

All the data file registers are copied to the clipboard both in text format and as OLE 2 data links. This data can be pasted into any other application as normal text using the "PASTE" command. If the application into which data is being pasted is OLE 2 aware (like EXCEL 5, WORD 6, etc), then "PASTE LINK" or "PASTE SPECIAL" commands can be used to hot link the data.

If the pasted data is linked, then any change in Data Ram in ICEPIC is automatically reflected in the pasted application.

Hot Key :

- **CTRL + C**

Paste (ALT + E P)

Inserts the text from the Windows Clipboard into the text editor windows at the current cursor position. Only text can be pasted.

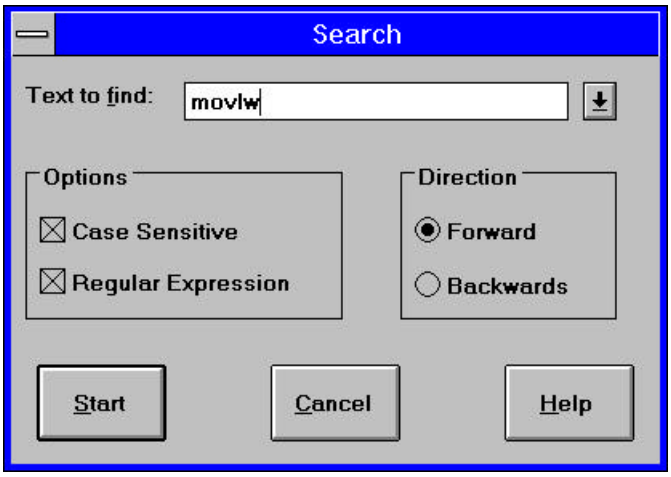
This function is available only in the text editor window(s).

Hot Key :

- **CTRL + V**

Find ... (ALT + E F)

Opens a dialog box to find the desired text. The text can be searched forwards or backwards through a file for a certain "pattern" of text. See Appendix A for details of search operations, metacharacters and "regular search expressions"

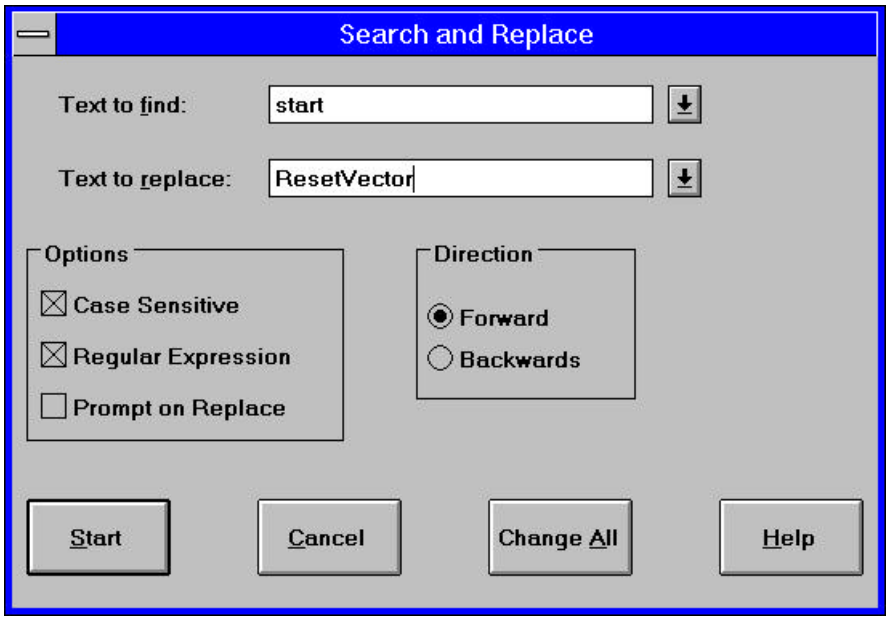


Find Next (ALT + E N)

Finds the next occurrence of the previously selected text using the Find menu command.

Replace (ALT + E R)

Opens a dialog box to find the desired text and replace it with the new text. See Appendix A for details of search operations, metacharacters and "regular search expressions".



Match Bracket (ALT + E B)

Matches brackets in the text editor window. If the character at the current position is one of the following :

{, }, (,), [,], ","

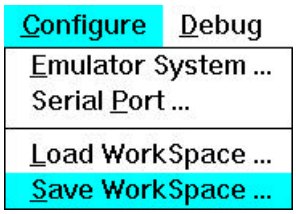
an attempt is made to find the matching brace or quote. If the matching pair is found, the current position is moved to the matched character. This feature is extremely useful while editing and debugging a "C" program.

Edit Tab Stops (ALT + E S)

Sets the length of the TAB in the editor. The default tab length is 8 spaces.

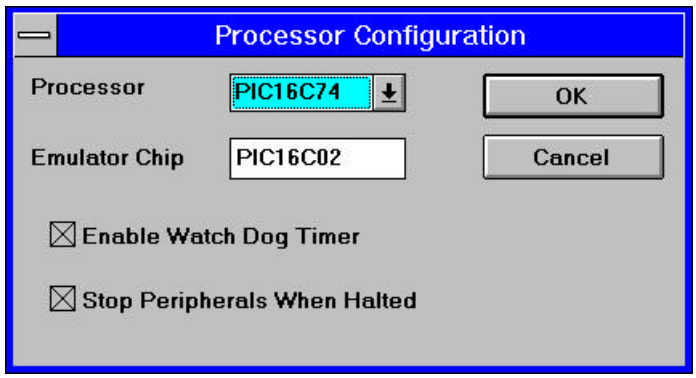
Configure Menu (ALT + C)

This menu has the following commands which are fully described below.



Emulator System ...

This menu command allows configuration of the emulator system. When selected, a dialog box as shown below is activated :



From the Processor drop down selection box, select the desired target processor. For example, in the PIC16C5x system, PIC16C54 through PIC16C58 can be selected. Note that some of the emulator systems support only one processor.

The Emulator Chip box is for information only and cannot be modified by the user. This box displays the type of the emulator chip used.

The Enable Watch Dog Timer selection check box enables or disables the Watch Dog Timer (WDT). Both PIC16C5x and PIC16Cxx have on chip WDT and is enabled as a Fuse Option in the actual processor. In the emulator system, it is implemented as a software selection option. Note that the WDT is reset when the processor is halted. It is also held in reset state in single step mode.

The "Stop Peripherals When Halted" option, freezes all the peripherals (like timers, PWM, etc) when the processor is halted. This option gives a snapshot of the processor's registers when it is halted. However, in some rare situations, the user may wish to continue to run the peripherals even when the processor is halted. For example, if the user is running a motor using PWM of PIC16C74, the PWMs will be stuck at either a 1 or 0 when the processor is halted. This may put the motor in an undesirable mode. To avoid this, uncheck this option and the PWMs will continue to run.

Warning (PIC16Cxx Processors) :

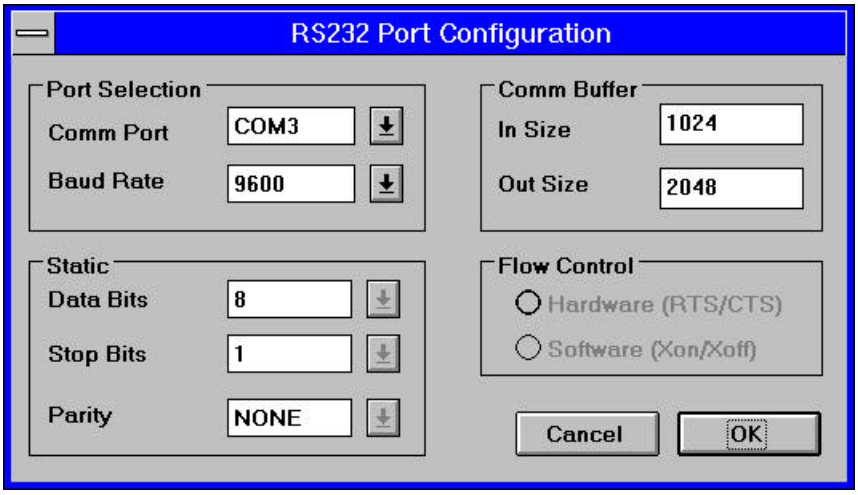
If the "Freeze" mode is enabled and if PORTB data is modified, the data of PORTB is not updated on the screen until a "Single Step" is executed. The data actually gets modified in the processor, but does not show up until the processor is out of halt mode for at least one cycle. This is not a bug in the ICEPIC system.

Advanced Feature

In PIC16Cxx series, the user can set break points on WDT overflow. See "Options Debug" menu.

Serial Port ...

This menu command allows the user to configure the serial port. The ICEPIC emulator system which connects to your computer via RS232 port, can be connected to any serial port of your system (provided MS Windows supports it) and the baud rate is software selectable. When this menu is selected a dialog box as shown below is opened:



This software under MS Windows allows the user only the first two options (namely Comm Port selection and Baud Rate selection). All other controls are for information only.

The "Comm Port" drop down selection box lists all the serial ports available under Windows. The user will be notified if the selected Comm port is already occupied by another application (e.g. serial mouse or a modem).

From the "Baud Rate" dropdown box, select the desired baud rate. Baud rate from 2400 bps to 57600 bps are supported. The ICEPIC system has an autobaud detect feature and no dip switches are necessary. Note that baud rates greater than 9600 bps may not be feasible on all computers. The highest baud rate achievable depends on many factors, like the speed of your computer, video driver, and most importantly whether or not you have a 16550 UART. See the Appendix D for more information

Recommendation

Connect the emulator system to either COM1 or COM2

Use serial I/O cards with 16550 UART and use Windows For Workgroups 3.11 to achieve high baud rates.

All other controls are not user selectable and are shown for information only.

Hot Key ALT +C P

What's Your UART ?

- At your DOS prompt (e.g. C:\), enter MSD.EXE
- Choose Comm Ports
- If your serial I/O card is equipped with old technology, the screen will show 8250 or 16C450
- If you have a high speed UART, the screen will show 16C550

Load Workspace ...

This menu command allows the user to select a previously saved configuration file from a standard file open dialog box. The default extension is WSP and this file must be generated by the ICEPIC software. When selected, the previously saved settings are restored.

Hot Key ALT + C L

Save Workspace ...

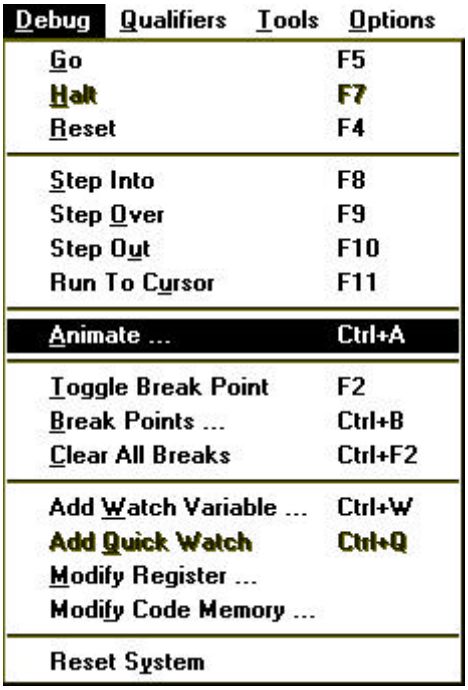
This menu allows the user to save the current workspace settings to a file, with a default extension of WSP. Various settings like, the window positions, colors, baud rate, comm port, debug options, etc are saved. These saved settings can be recalled later, by the "Load Workspace ..." menu command.

Note that these settings are automatically saved to the default ICEPIC.INI file under the WINDOWS directory when one exits from the program. Use this menu if there is a need to save multiple configurations. The ICEPIC.INI is automatically read when a new session is begun.

Hot Key ALT + C S

Debug Menu (ALT + D)

The debug menu has the following commands which are described below.



Go (ALT + D G)

This menu command starts the processor running in real time. The processor continues to run in real time until either a break point is encountered or is interrupted by a user halt command.

Processor running in real time is indicated both in the Status Bar ("Running" test is displayed) and in the Tool Bar (a rotating wheel is displayed).

NOTE : The WDT is activated if enabled.

This command is disabled if the processor is already running in real time.

NOTE : The processor will remain in reset state if the MCLR pin is held low, or left floating. Use the MCLR jumper link (J3 on the daughter board) to pull up MCLR to a high state to overcome this situation.

Hot Key :

- **F5**
- **Use the toolbar button**

Halt (ALT + D H)

Halts the processor if running in real time. This menu is enabled only if the processor in running in real time. After halting the processor, it's status is read and windows are updated with the new status of the processor.

NOTE : If the processor is in "SLEEP" state, HALT is not recognized by the emulator system, and the user must reset the processor to halt or come out of sleep mode.

Hot Key :

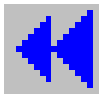
- **F7**
- **Use the Toolbar button**

Reset (ALT + D R)

Resets the processor by toggling the MCLR pin to the emulator chip. The MCLR pin from the user's target is not disturbed. The emulator system interprets the Reset command in two different ways :

- If the processor is already in HALT state, the processor is reset, and put in halt state at the reset vector.
- If the processor is running and the user presses the Reset button, MCLR pin is activated for a few microseconds and then deactivated.

Hot Key : F4

Use the Toolbar button 

Step Into (ALT + D S)

Single step the processor, essentially executing the instruction at the current program counter (halted location). After executing the instruction, the processor's new status is read and updated information is shown in all windows.

Hot Key :

- **F8**
- **Use the toolbar button**

Step Over (ALT + D O)

This function executes the subroutine (function call). If the current instruction is a CALL, the processor will halt. If the instruction at the current location is not a CALL, then a normal single step function (Step Into) is executed.

Use this function to check the flow of the program and avoid going into an already debugged subroutine.

Hot Key :

- **F9**
- **Use the toolbar button**

Step Out (ALT + D U)

When executed, this function runs the processor until it returns from the current subroutine (function call). In other words, the processor comes out of the innermost function call. If the processor is not in any function call, then this command is treated the same as a normal single step (Step Into) function.

Hot Key :

- **F10**
- **Use the toolbar button**

Run To Cursor (ALT + D N)

When executed, thus function runs the processor as described below :

In Editor : Runs the processor until the cursor position.

In Code Memory Window : Runs the processor until the highlighted location.

This menu function is disabled when none of the above window(s) are inactive.

Hot Key :

- **F11**
- **Use the toolbar button**

Animate (ALT + D A)

This function is essentially a continuous single step mode. The processor executes one instruction at a time continuously until halted by the user. In essence, it is a slow motion execution of the processor.

NOTE : By default, after each instruction, all registers are read and updated. To improve the speed, uncheck the "Read Register in Animation" checkbox in the Options Debug menu.

Use the HALT tool bar button to stop the animation mode.

Advanced : By default, each single step is executed approx. every 500 mSecs, using an internal timer. This should be adequate time for most of the computers at all the baud rates supported. You may fine tune this speed to your requirements (for faster response) by modifying the "ICEPIC.INI" file stored in your Windows directory. Edit the [Animate] section and change the Timer = to the desired value in mSecs.

Toggle Breakpoint (ALT + D T)

This function sets or clears the breakpoint at the currently selected location in the code memory window. A location with break point set is shown in RED. These break points are also available from the "Break Points ..." menu command.

The Right Mouse button may be used in the code memory window to obtain a floating pop-up menu to set break points.

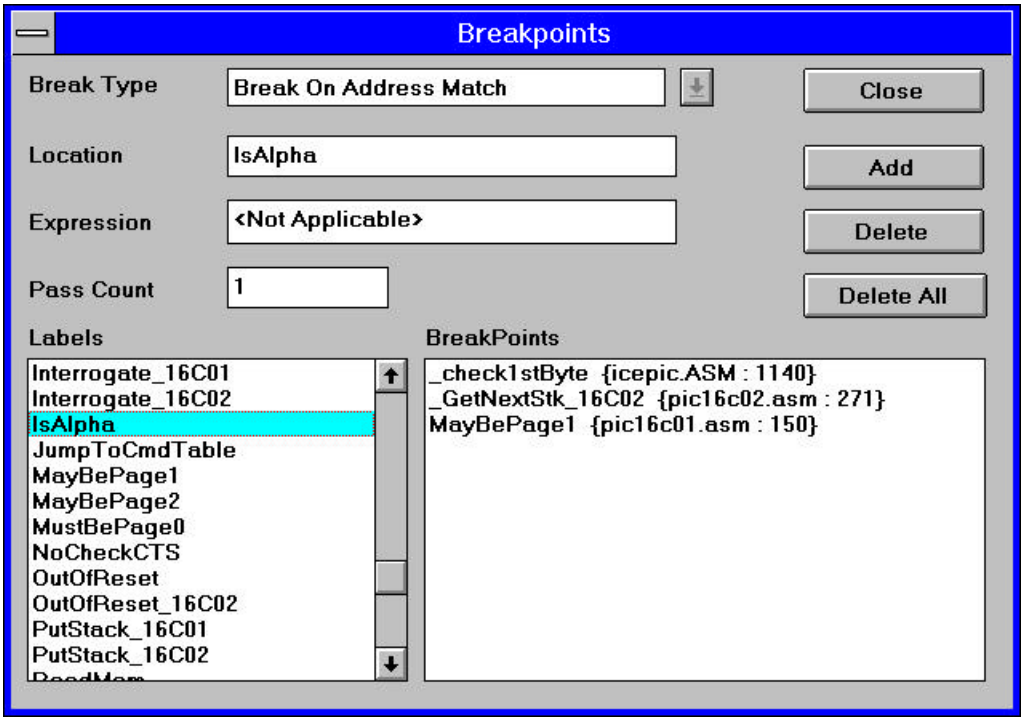
This function is only available if the Code Memory Window is active.

Hot Key :

- **F2**
- **Use the toolbar button**

Break Points ... (ALT + D B)

When this function is selected, a dialog box as shown below, is opened. It is used to manipulate various breakpoint settings.



Break Type

In the ICEPIC emulator system, only "break On Address Match" is implemented. Real time breakpoints can be set only on program memory address match condition.

Location

Break point address entered in this location is used to set break points when the Add button is pressed. The break point location can be entered as a valid Label or as a HEX value or as a DECIMAL value. See Appendix “Data Types and Formats” for the syntax of HEX & DECIMAL values.

Expression

Not implemented in the ICEPIC emulator system

Pass Count

In the ICEPIC emulator system, Pass Count is always set to 1. This means, the processor is halted the first time a breakpoint is encountered.

Labels

This list box displays all the current address "Labels" used in the source code. These symbols are read from the COD file. If a HEX file is used, these symbols are unavailable.

To add a break point at one of these locations, select the desired address label and click on the ADD button or simply double click on the desired address label.

BreakPoints

This list box displays the current active breakpoints. To remove a breakpoint, select the desired item and click on the Delete button or simply double click the item to be deleted.

Add Button

A breakpoint is set and added to the breakpoint list at the address entered in the "Location" box.

Delete Button

Deletes the breakpoint at the address entered in the "Location" box

Delete All Button

Clears all breakpoints.

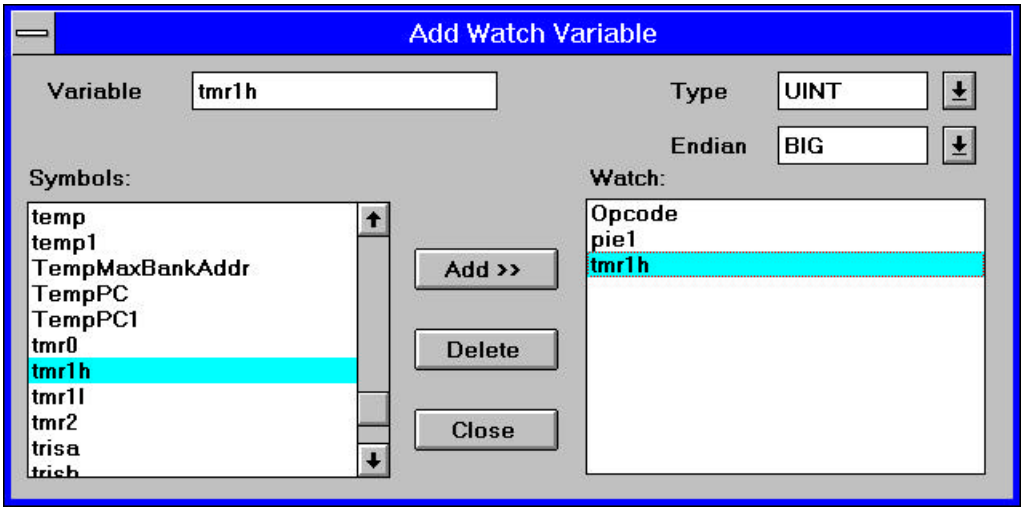
Hot Key : Ctrl + B

Clear All Breaks (ALT + D C)

Removes all break points, the code memory window is updated.

Add Watch Variable (ALT + D W)

This menu command enables variables to be entered into the watch window. When selected, this menu brings up a dialog box as shown below



Variable

This is a text field in which the variable address (or a valid symbol) must be entered. The variable address can be any valid Data File Register of the target processor. If symbolic information is available, symbols are shown in the "Symbols" list box.

To add this variable to the watch window, click on the "Add" button

Symbols

This listbox displays all the currently loaded symbols (except code address labels). A watch point can be set on any of these symbols. To set a watchpoint select the desired symbol and click on the "Add" button or simply double click the symbol.

Watch

This list box displays all the currently active watchpoints. To delete a particular watch item, select the item to be deleted and click on the Delete button or simply double click the item to be deleted.

Type

The ICEPIC user interface supports data type of BYTE, CHAR, INT, UINT, LONG and ULONG. These data types are described in detail in Appendix E. Select the desired type of data before adding a watchpoint.

Both PIC16C5x and PIC16Cxx are 8 bit processors. All data file registers are 8 bit wide. However, if multibyte is used, higher precision arithmetic can be implemented in software. In this situation, it may be desirable for the user to watch data of multiple bytes as a single variable.

For example, in PIC16C74, the peripheral Timer 1 is a 16 Bit Timer with low byte implemented at address "tmr1l" and high byte at address "tmr1h" (tmr1l +1). Instead of watching (and modifying) them as separate variables, the user can add "tmr1l" as a 16 Bit unsigned value (UINT).

Endian

ICEPIC user interface supports watching and modifying data file registers of multibyte length. Since this is essentially a software implementation, the user interface needs to know if the high byte is at a lower or higher address from the low byte.

In Big Endian mode, high byte is at a higher address than the low byte, whereas in the Little Endian mode, high byte is at a lower address than the low byte.

Hot Key :

- **Ctrl + W**
- **Use the toolbar button**

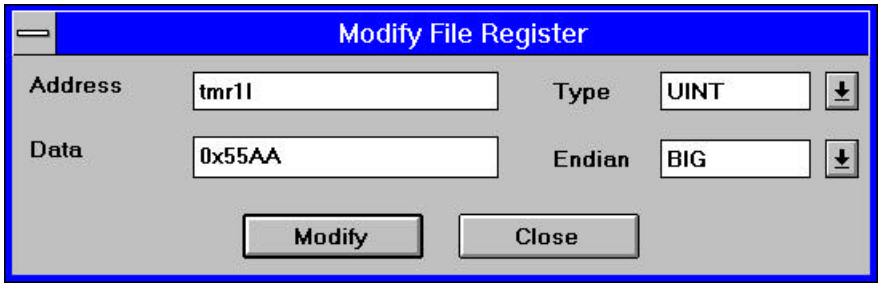
Add Quick Watch (ALT + D Q)

The current word under the cursor in the text editor window is added to Watch Window as a variable. If a COD file is loaded, then the added variable's data type is automatically selected and represented in the Watch Window.

For example, if the current word in the editor is "Motor_Speed" and it is declared as "unsigned long" in the C-program, then the "Motor_Speed" variable is added to Watch Window and is represented as a 2 Byte value.

Modify Register ... (ALT + D M)

This menu command allows modification of a data file register (including Special Function Registers) and when selected, brings up a dialog box as shown

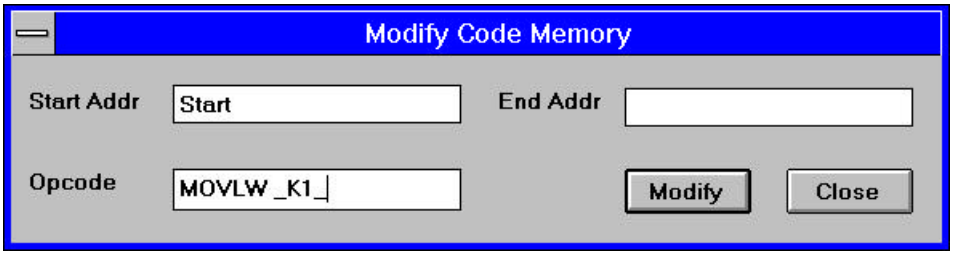


In the "Address" edit box, type in the address of the desired data file registers. The address may be entered symbolically. In the data window, enter the new data and click on the "Modify" button to modify the register. The data can be a multibyte value depending on the selection of "Type" and "Endian".

For more information on the multibyte values and data types, see Appendix E.

Modify Code Memory ... (ALT + D F)

This menu command allows the modification of a program memory location or to 'fill' the code memory with the desired opcode.



To modify a location, enter the desired code memory address in the "Start Addr". The address can be entered as a number (HEX or DECIMAL) or as a label if symbolic information is loaded. In the "Opcode" editbox, enter the desired opcode value of a valid assembly language pnemonic and click on the "Modify" button. ICEPIC user interface has an onboard single line assembler and a disassembler.

If only one location is to be modified, the user need not enter any data in the "End Addr". If a range of locations is to be filled with an opcode, enter the end address in the "End Addr".

Tip : To modify a code memory location, *double click* the desired location in the code memory window.

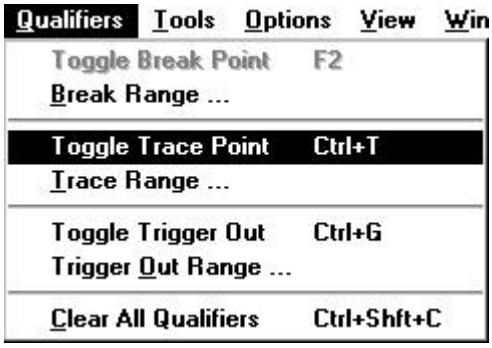
Reset System (ALT + D Y)

This menu command performs a system wide reset. This function must be used only on error conditions and should not be used to reset the target processor. This command puts the complete emulator system in a power-up state.

Use this function if serial errors occur or if the emulator system hardware goes through a power-up sequence.

Qualifiers Menu (ALT + Q)

The following commands are available under this menu



This menu contains functions which set/clear various qualifiers, namely break points, trace points & trigger output points. Not all the above mentioned qualifier settings are available in all systems. For example, ICEPIC-II systems has all the above features, but ICEPIC system does not support Trace & Trigger points.

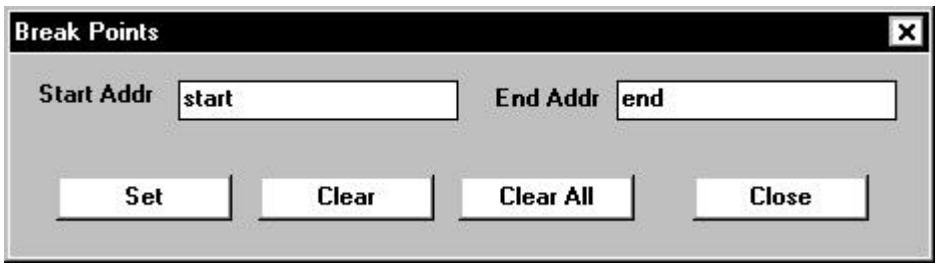
Toggle Break Point (ALT + T B)

Same function as mentioned under “Debug Menu”.

Break Range (ALT + Q B)

Set or clear breakpoints with an address range. The End address must not be less than the Start address. Symbolic address labels can be entered.

This command is applicable only for ICEPIC-II model.



Toggle Trace Point (ALT + Q A)

This function sets or clears the trace point at the currently selected location in the code memory window. A location with trace point set is shown with a special icon.

This function is only available if the Code Memory Window is active.

This command is applicable only for ICEPIC-II model.

Hot Key :

- **Ctrl + T**

Trace Range (ALT + Q T)

Set or clear tracepoints with an address range. The End address must not be less than the Start address. Symbolic address labels can be entered.

This command is applicable only for ICEPIC-II model.

Toggle Trigger Out (ALT + Q G)

This function sets or clears the trigger out point at the currently selected location in the code memory window. A location with a trigger out point set is shown with a special icon.

This function is only available if the Code Memory Window is active.

This command is applicable only for ICEPIC-II model.

Hot Key :

- **Ctrl + G**

Trigger Out Range (ALT + Q O)

Set or clear tracepoints with an address range. The End address must not be less than the Start address. Symbolic address labels can be entered.

This command is applicable only for ICEPIC-II model.

Clear All Qualifiers (ALT + Q C)

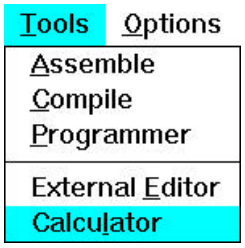
Clears out all qualifier points, i.e., removes all break, trace & trigger output points.

Hot Key :

- **Ctrl + Shift + C**

Tools Menu (ALT + T)

The following commands are available under this menu



This menu contains functions which launch various tools. Before being able to execute these tools, they must first be setup in the “Options Tools” menu command.

Assemble (ALT + T A)

Executes the assembler program. By default, MPASM assembler is used (runs under a DOS Window). To assemble a program, edit the "Option Tools" menu and add the command line parameter to the Assembler section.

Compile (ALT + T C)

Executes the C Compiler program. By default, MPC compiler from Byte Craft is used (runs under a DOS Window). To compile a program edit the "Option Tools" menu and add the command line parameter to the Compiler section.

Programmer (ALT + T P)

Executes the programmer. By default, Microchip PROMATE is used (runs under a DOS Window). To specify another programmer edit the "Option Tools" menu and add the command line parameter to the programmer section.

Editor (ALT + T E)

Executes the editor program. By default, Notepad program (supplied standard with MS Windows) is used as the editor. To use your favorite editor, modify the "Editor" field in "Options Tools" menu.

Calculator (ALT + T L)

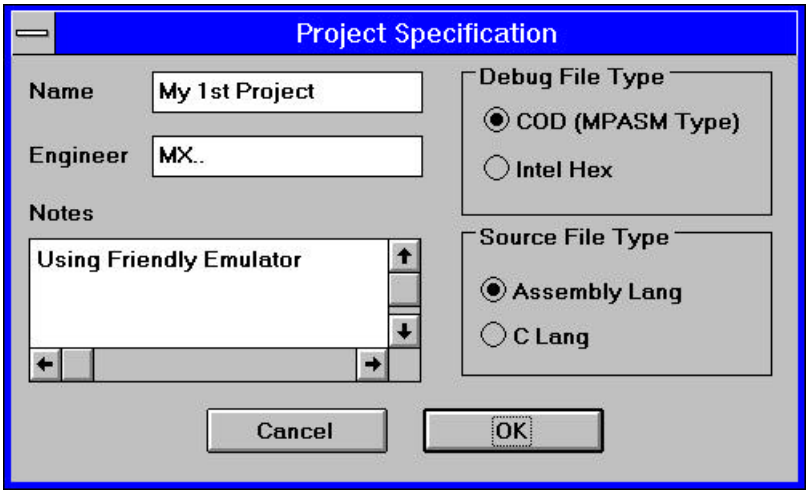
Runs the standard Windows calculator. This calculator has a scientific mode and may prove to be useful for HEX conversions, Left shifts, etc.

Options Menu (ALT + O)

This menu contains the following commands. It enables various options of the emulator functions and the user interface functions to be set.

Project... (ALT + O P)

When selected, this menu command brings up a dialog box as shown below. Use of this box allows various options of the current project to be saved. These options do not affect the emulator system hardware functions.



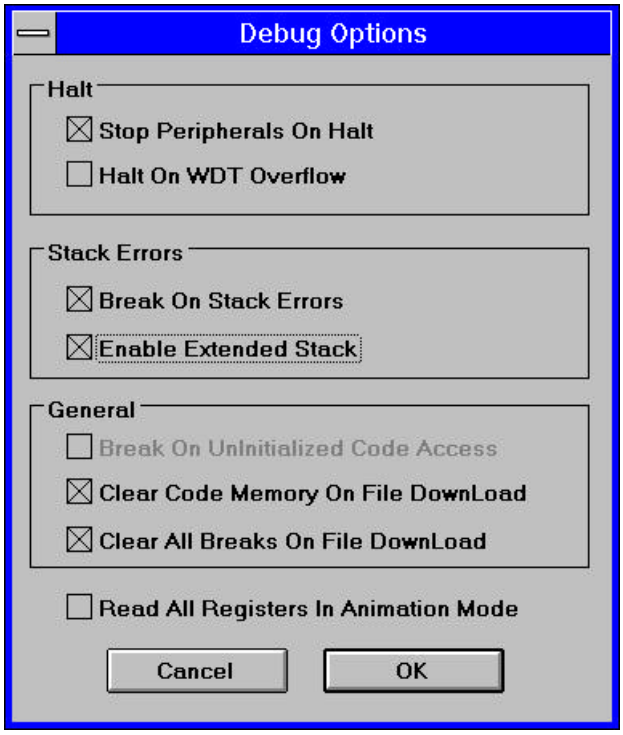
In the "Name" field, enter the name of your project. In the "Engineer" field, you may enter your name. The "Notes" field can be used to enter brief notes of your project. For example, you may want to enter the Version Number and any changes you have made.

The "Debug File Type", has two options, "COD" and "Intel Hex". It is used to select the default extension of the object files. Use of the COD option is recommended.

The "Source File Type" sets the default type of the source code. Currently only the Assembly Language is used.

Debug... (ALT + O D)

The debug options menu contains various emulator debug functions. These functions affect the hardware functions and are described in detail below:



Stop Peripherals On Halt

Check this box if the peripherals (like timers, PWMs, etc) are to be frozen when the processor is halted. This option can also be set from the "Configure Emulator" menu. This option gives a snapshot of the processor's registers when it is halted. However, in some rare situations, the user may wish to continue to run the peripherals even when the processor is halted. For example, if the user is running a motor using PWM of PIC16C74, the PWMs will be stuck at either a 1 or 0 when the processor is halted. This may put the motor in an undesirable mode. To avoid this, uncheck this option and the PWMs will continue to run.

PIC16C5x	This option is always turned on, and cannot be disabled
PIC16Cxx	User selectable

Warning (PIC16Cxx Processors) :

If the "Freeze" mode is enabled, and if PORTB data is modified, the data of PORTB is not updated on the screen until a "Single Step" is executed. The data actually gets modified in the processor, but does not show up until the processor is out of halt mode for at least one cycle. This is not a bug in the ICEPIC system.

Halt On WDT Overflow

Checking this option will halt the processor when the Watch Dog Timer (WDT) overflows. In a typical situation, where WDT is used, the user resets the WDT by using the CLRWDT instruction. Often during program development cycle, the user may forget to place the CLRWDT instruction at the appropriate place, thereby overflowing the WDT and resetting the processor. Selecting this option will halt the processor, informing him that WDT has overflowed.

Uncheck this option, if WDT overflow is desired for your application.

PIC16C5x	This feature is not available
PIC16Cxx	All processor's of this series have this feature.

Break On Stack Errors

This option is one of the most useful debug features of this emulator system. The PIC16Cxx series have a lot of interrupts (more than 10 in PIC16C74), but has only 8 stack levels. Often, stack overflow occurs during code development stages, it is then extremely hard to find the location of the problem.

When checked, the processor halts and informs the user of the errors. These errors can be either a stack overflow or underflow. Stack overflow is a situation in which the user tries to access more than the 8 levels of PIC16Cxx stack. Stack underflow takes place when a pop occurs (return, retlw, retfie), but the stack is empty.

PIC16C5x	This feature is not available

See "Enable Extended Stack" function for more information on debugging stack errors.

Enable Extended Stack

This function applies to PIC16Cxx series only. The PIC16Cxx series has 8 stack Levels. In some situations, when developing firmware etc, the stack is overflowed. This function (Enable Extended Stack) along with the "Break On Stack Errors" helps to debug these situations.

The emulator system for all processors of PIC16Cxx series, has the ability to extend the stack from 8 levels to 12 levels when this option is selected. When stack errors occur, the flow of the program can still be viewed, allowing the cause of the overflow to be located.

See "Break On Stack Errors " function for more information on debugging stack errors.

Break On Uninitialised Code Access

This function is currently unavailable. When implemented, a break point occurs when your code falls into a location that has not been initialised.

Clear Code Memory On File Download

Check this box to clear all memory to erased EPROM state whenever a new object code is downloaded. If unchecked, the current code is modified only if the new object code uses those locations. This option is set by default.

Clear All Break On File Download

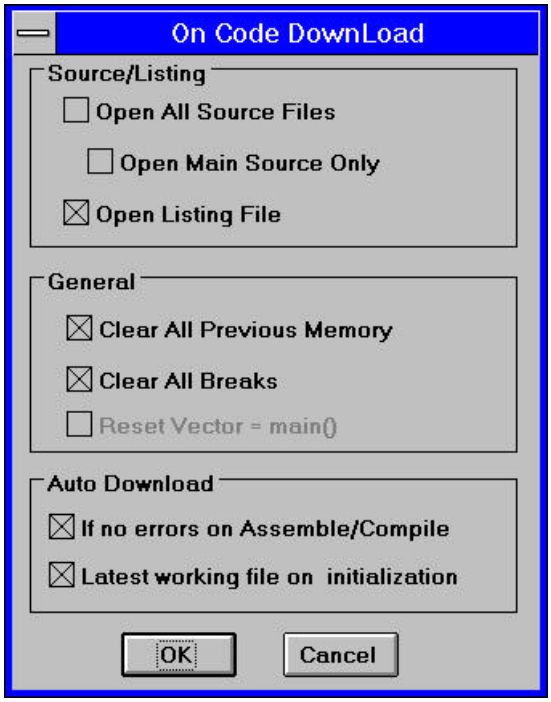
Checking this option will remove all breakpoints whenever a new file is downloaded. This option is set by default.

Read All Registers In Animation Mode

When the emulator system is in animation mode (see "Debug Animate" function for more details), all registers of the processor are read after executing each instruction. This slows down the process. If unchecked, only the minimal status of the processor (PC, STATUS, etc) is read while the processor is in animation mode. This is the most efficient mode.

Code Download ... (ALT + O L)

This menu command brings up a dialog box as shown below. Use this dialog box to perform appropriate actions to take place upon a new code download.



Open All Source Files

Opens all source files in the current project when the COD File is downloaded successfully. For example, if the project "main.asm" has other include files ("include1.asm"), then both "main.asm" and "include1.asm" are opened for source level debugging.

Open Main Source Only

When a COD file is downloaded successfully, then open main source file only for source level debugging. For example, if a COD file of "Proj.COD" is downloaded, then open only the main source file "Proj.asm" for source level debug even if "proj.asm" has other include files. This option is especially useful if there is only one source file and many other include header files.

Open Listing File

Opens the listing file upon a successful download of a COD File. This is the recommended way. Source debugging using the list file in a C-Program (using MPC) essentially gives a mixed code debug.

Note that the listing opened as a part of the project cannot be edited and is Read Only.

Clear All Previous Memory

If this option is checked, the code memory is cleared to 0xFFFF (erased EPROM state) before downloading new code. If unchecked, the new code replaces the old code, but does not clear other memory locations.

Clear All Breaks

If this option is checked, all break points are cleared when ever a file is downloaded to the emulator. If unchecked, the current break points remain active.

Reset Vector = main()

If checked, upon download of a "C" code (from COD file generated by mpc), the reset vector is set to "main()". This option does not apply to MPASM (assembler) generated code.

The main entry to any "C" program is "main()". This is the beginning of the program to a user. However this is NOT the Reset Vector of the processor. The "C" compiler generates some initialization code starting from the processor's reset vector before it jumps to the user "main()". The ICEPIC software, upon download of a "C" code, runs until "main()" so that the user sees it as the beginning of his/her program.

Turn off this option if you desire the ICEPIC to start from the Processor's Reset Vector. You may also need to turn off this option if target system pulls down the MCLR pin, thus not enabling ICEPIC to run until "main()'.

CAUTION

If this Option is enabled, make sure the MCLR is NOT held low before you download your code.

If No errors on Assemble/Compile

Automatically downloads code (loads the COD File) to the emulator upon successful Assembly or Compilation if no errors are found in the Error File (.ERR file generated by either Assembler, MPASM/MPASMWIN or by the C-Compiler MPC). Make sure the Error File Generation is turned on in the assembler options. This function is operable with either MPASMWIN or MPASM for DOS.

Latest Working File on initialization

Check this option if you want the ICEPIC to download the working file that was previously in use when ever you start your ICEPIC software. This is equivalent to starting ICEPIC & then downloading the most recently used (MRU) file from the File Menu.

Real Time Trace... (ALT + O R)

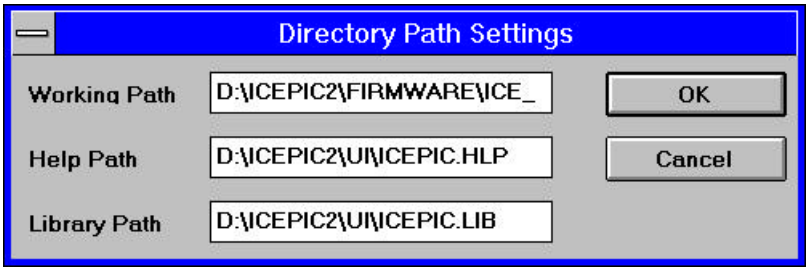
Using this command the user can select the Option to capture the dummy/2nd cycle cycle(s) into the Trace buffer. Some of the PIC16Cxx Microcontrollers are executed in 2 Cycles (whenever there is an address jump, like CALL, GOTO, etc). Selecting the Option to capture the 2nd cycle, captures the address bus during this 2nd cycle.

This option to trace the 2nd cycle is most useful for counting the total number of cycles executed.

This command is applicable only for ICEPIC-II model.

Directories... (ALT + O I)

This function allows the default path of various directories to be set. It should be used when debugging a C Language program.



The "Working Path" contains the path of the source files. The "Help Path" contains the path of the online help files. The "Library Path" contains the path of the library files.

Colour... (ALT + O C)

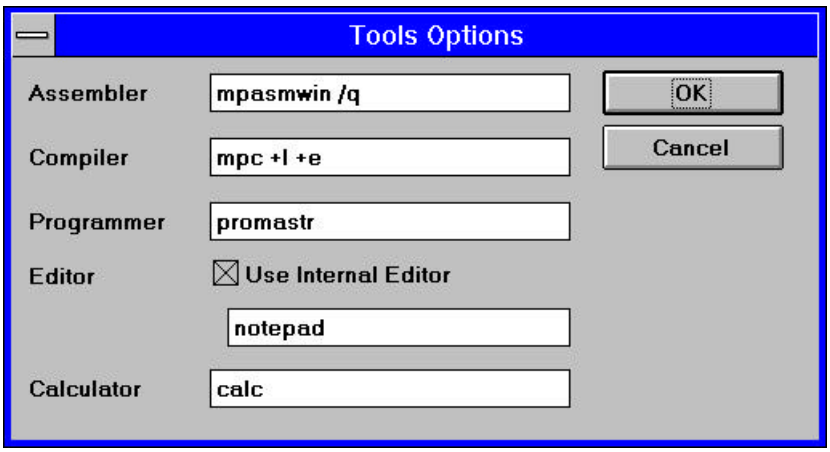
Colours of both text and background for all windows can be customized for the ICEPIC. To set the colors of each window, click the right mouse button in the desired window and select the colour option from the floating menu.

Font ... (ALT + O F)

Currently unimplemented. All fonts are currently of fixed length and use the system font.

Tools ... (ALT + O T)

This menu command allows modification to the default tool settings.



All the new settings are automatically saved and restored every time a new emulation session is started.

Assembler

Enter the desired assembler name. The default assembler is "mpasm.exe". In addition to the executable file name, command line parameters may be added.

A typical example is : MPASM myproj.asm /L

Compiler

Enter the compiler name. By default, "mpc" is used. "MPC.EXE" is the C compiler from Byte Craft. Any other compiler may be used, but note that only the object code of COD format is supported.

Programmer

The default programmer is "PROMATE", the universal programmer from Microchip Technology. You may modify this setting for any other programmer.

Editor

By default, the built in editor is used and is the recommended way. However an external editor can be used by unchecking the "Use Internal Editor". Note that using an external editor does not give you source level debugging right from the editor. Source level debugging can be performed using the Listing File loaded into the ICEPIC even if the source is being edited by an external editor.

Calculator

Enter the desired calculator name. For most situations the default calculator (calc.exe) that comes with MS Windows is quite good. Make sure to select the scientific mode from the calculator program to get the most useful functions while debugging or writing assembly code.

View Menu (ALT + V)

This menu contains various view commands, to open or show all child windows. These menu commands are explained below:

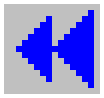
Toolbar (ALT + V L)

Shows or hides the main toolbar button. The main toolbar button contains the most commonly used menu commands.

Execute the File Download command

Save the current file

Assemble the current file


Reset the target processor

Execute the GO command

Execute HALT command

Execute Step Into command

Execute Step Over command

Execute Step Out of command

Execute Toggle Breakpoint command

Execute Add Watch Variable command

Print the current file

Execute the About Box command

Put the system in a context sensitive help mode.

Enables or disables all break points

Status Bar (ALT + V B)

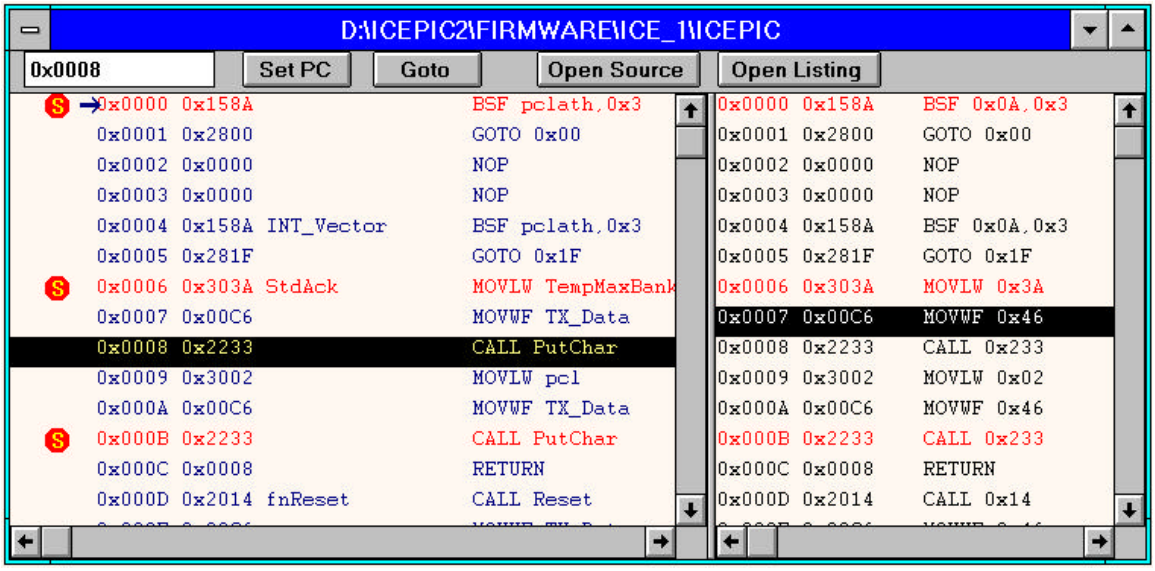
Shows or hides the status bar. The status bar is shown at the bottom of the main window. The status bar displays various information, including menu descriptions, current processor, current program counter, etc.

Note that the first pane also displays a brief help on each command. For example, select a menu and watch the help string being displayed in the status bar.

Code Memory (ALT + V C)

Opens the Code Memory (Program memory) window. This window displays the downloaded code memory, current break points and the current program counter location. The user can set or clear breakpoints directly from this window. The code window has two views, one showing the source code and the other showing the disassembled code.

Source code from the user source file is displayed if the default COD file is used. If no source code information is available, disassembled code is displayed. A typical code window appears below



The view on the left side displays the source code (if available) and the view on the right hand side displays the disassembled code. The "solid bar" separating the two views can be dragged either left or right to adjust the size.

To set or clear a breakpoint at a desired location, select the desired line and press F2 or use the toggle breakpoint toolbar button. The breakpoints are shown coloured Red.

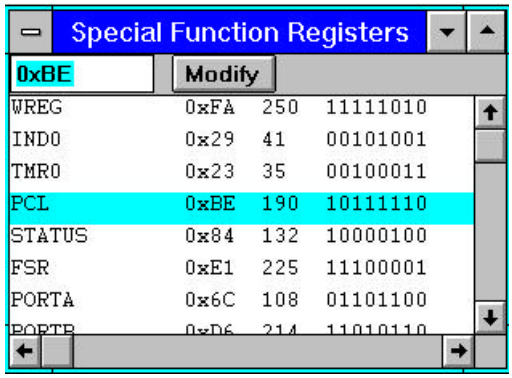
The current program counter is displayed with ">>" at left end of the window (left of the address field).

Various other functions specific to these views can be accessed by using the floating menu. The floating pop-up menu is activated by clicking the Right Mouse Button in this window. These menu items are explained below

The "Open Source File" menu command opens the source code file by running the editor. The source file opened, depends on the currently selected location in the code memory window. The Background and Text Color menus allow the colours to be set. All other menus are described elsewhere in this chapter.

SFR (ALT + V R)

This window displays the Special Function Register's names and their HEX & DECIMAL values. The value of the register can be modified by selecting the desired SFR, entering the new value in the edit box and then pressing the “Modify” button. A typical view of the SFR is shown below.

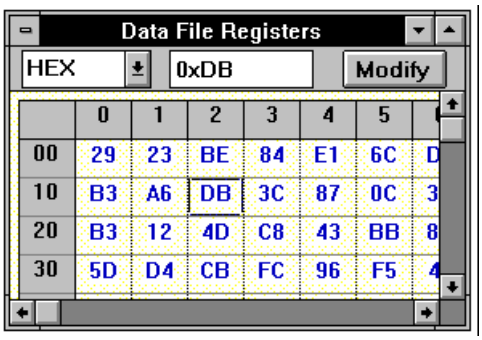


As can be seen, both HEX & DECIMAL values are displayed for each Special Function Register. The SFR names used are the same as those described in the data sheet of the processor being emulated. Some registers like W Register (WREG) are also shown in the SFR window, even though these are unmapped into the data file register array.

TIP *Use the Right Mouse Button to access a floating menu to set colors.*

Data File Registers (ALT + V D)

This window displays all the data file registers, including SFRs (mapped into Data File Register array). The number of registers displayed depends on the processor being emulated. A typical view of this window is shown below (for PIC 16C74).



All data is represented in the form of a tabular grid, similar to a spreadsheet. From the drop down list box, the user may choose to display the data in either HEX or DECIMAL format.

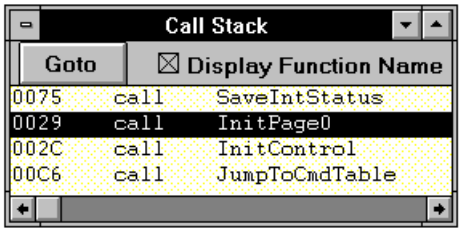
To modify a register value, simply select the desired cell, enter a new value in the edit box and click on the “Modify” button.

NOTE If a particular address is unimplemented in the processor, the data is shown as "X"

TIP *Use the Right Mouse Button to access a floating menu to set colors.*

Call Stack (ALT + V S)

The Call Stack window displays the stack of the processor. A typical view of the stack window is shown below



Checking the "Display Function Name" displays the function call corresponding to each stack value. Clicking on the GOTO button will bring the function into view, if the Code Memory Window is opened.

PIC16C5x

Always displays 2 levels of stack, irrespective of whether the stack is used or not.

PIC16Cxx

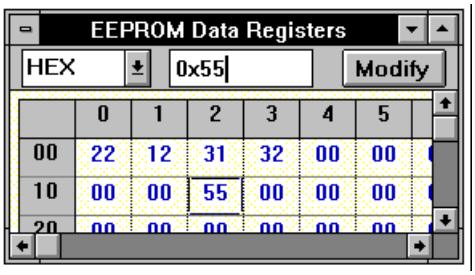
The number of stack levels displayed match the number of CALLs made. If stack overflow or underflow (stack errors) occurs, and if extended stack is selected, stack

levels up to 12 levels are displayed. For further information on stack errors, see Break On Stack Errors & Enable Extended Stack.

To view the corresponding stack location in the code memory window, simply select the desired stack location and press the "Goto" button

EEPROM Data (ALT + V E)

This window is enabled only if the target processor has on-chip EEPROM data memory (like the PIC16C84). The data is shown in the window in the form of a tabular grid, much like a spreadsheet.



The data can be watched and modified either in HEX or DECIMAL formats. To modify a certain location, select the desired cell (click the left mouse button), enter the desired new value and click the “Modify” button.

Real Time Trace (ALT + V T)

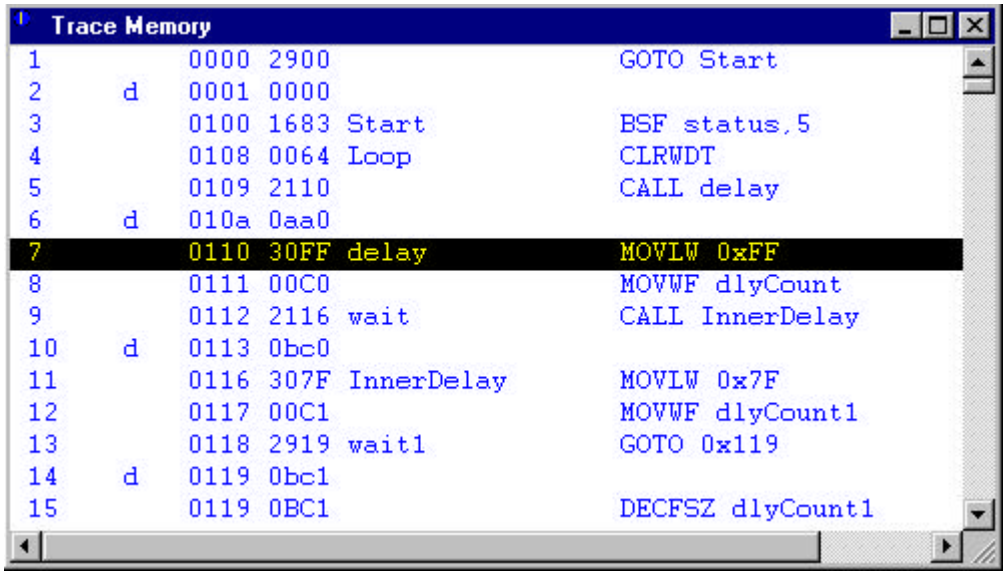
This window is enabled only if your system supports the Real Time Trace Option. Currently ICEPIC-II has this feature. This window displays the data in trace buffer as shown below.

In the hardware, only the address bus is captured. The Opcodes are shown as corresponding to the data at the captured address. The data at the corresponding address is displayed using the Listing File if the list file is loaded. Otherwise the data is disassembled.

If the 2nd Cycle capture option is used, the data is displayed without any disassembly, and is indicated by d in the 2nd column.

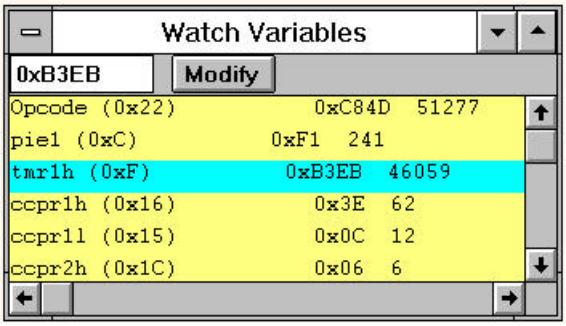
NOTE : Due to certain limitations, the trace buffer displays only 4095 locations (instead of 4096) when the trace buffer is overflowed.

A sample display of the trace window is shown below:


Trace Memory
1 0000 2900 GOTO Start
2 d 0001 0000
3 0100 1683 Start BSF status,5
4 0108 0064 Loop CLRWDT
5 0109 2110 CALL delay
6 d 010a 0aa0
7 0110 30FF delay MOVLW 0xFF
8 0111 00C0 MOVWF dlyCount
9 0112 2116 wait CALL InnerDelay
10 d 0113 0bc0
11 0116 307F InnerDelay MOVLW 0x7F
12 0117 00C1 MOVWF dlyCount1
13 0118 2919 wait1 GOTO 0x119
14 d 0119 0bc1
15 0119 0BC1 DECFSZ dlyCount1

Watch Window (ALT + V W)

This window is used for watching data file registers of user's choice. The user can watch and modify the desired registers symbolically. A typical view of the watch window is shown below:



Watch variables can be set using data types of BYTE, CHAR, INT, UINT, LONG and ULONG. The multibyte data values can be set in either Endian mode. For example, notice the "tmr1l" register shown as an UINT value.

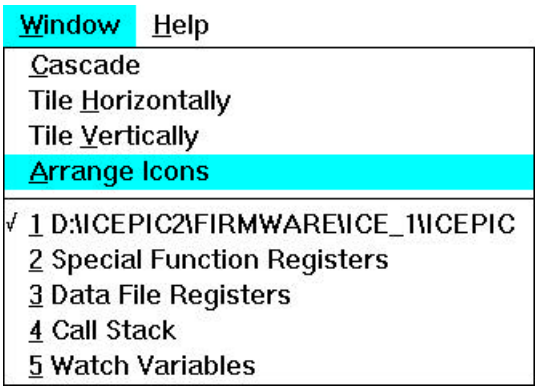
To modify a certain watch register, select the watch point, enter a new value (either HEX or DECIMAL) and click on the “Modify” button.

Debug Window (ALT + V G)

This window is currently disabled. It is intended for future use as an advanced debug feature.

Window Menu (ALT + W)

This menu command allows the user to access functions that arrange the child windows.



Cascade (ALT + W C)

Arranges all opened windows by cascading them. It is a Windows operating system related function and is not specific to the emulator.

Tile Horizontally (ALT + W H)

Arranges all windows by tiling them horizontally

Tile Vertically (ALT + W V)

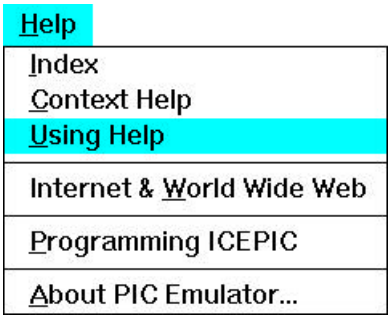
Arranges all windows by tiling them vertically

Arrange Icons

Arranges all the icons, if available.

Help Menu (ALT + H)

This menu allows access to the help commands and the information box.



Index (ALT + H I)

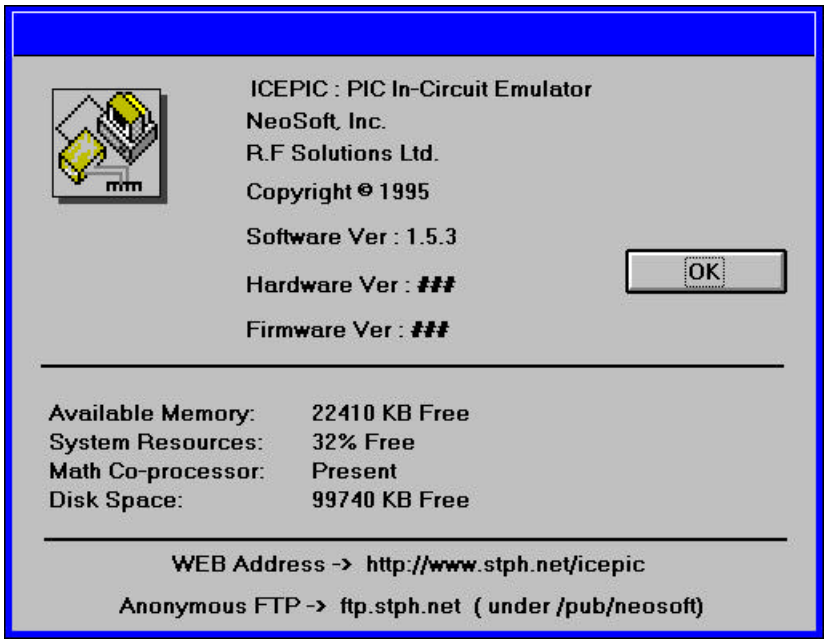
Opens the ICEPIC On Line help file and brings the index into view.

Using Help (ALT + H U)

Opens the standard Windows Help system and explains how to use the Windows On Line Help system.

About ICEPIC Emulator (ALT + H A)

This menu command will pop-up a dialog box as shown below. This dialog box displays the copyright information, hardware revision number, firmware version number, software version number and other system information.



Use the version numbers shown in this dialog box to see if you have the latest software and firmware.

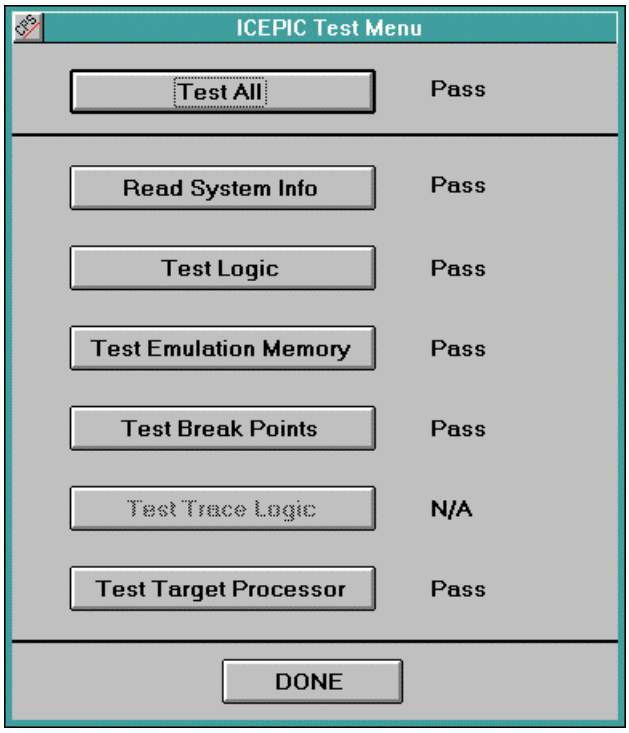
Chapter 4: Diagnostics

NOTE

Before running the diagnostics program the user must follow these rules:

- Do not connect the ICEPIC system to the target application board.
- The MCLR pull-up jumper (Link3, LK3) must be connected
- Clock and power must be present.

A separate Windows executable program ("icediags.exe") is installed by the installation program onto your hard disk. The installation program sets the command line parameters for this program according to your choice of the Comm port during the installation procedure. To run this diagnostics program, simply double click the "icediags" icon in the "PIC16CXX Tools" program group. When executed, a dialog box as shown below appears :



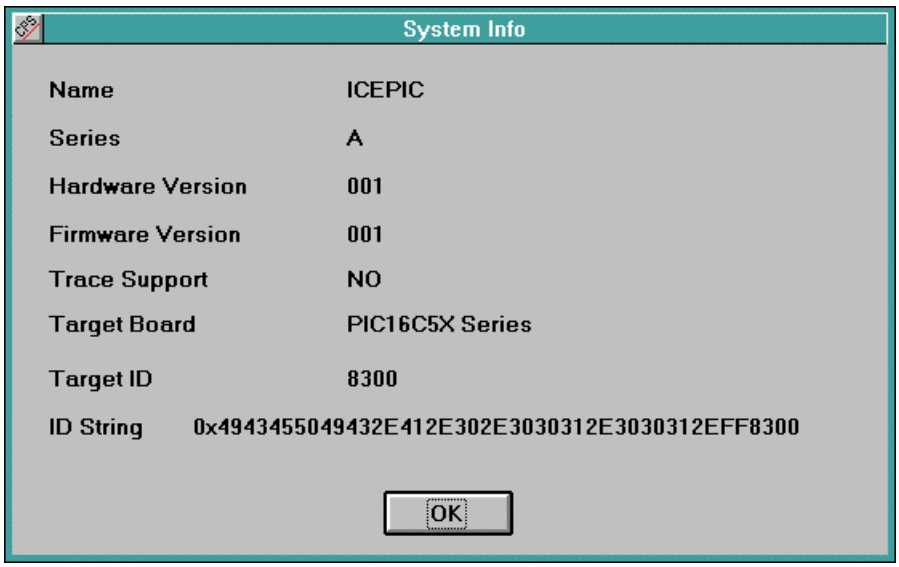
As shown in the above dialog, click on the "Test All" button to test all functions. Each block may be tested individually by clicking the desired test button. If a function block does not pass the test, failure is displayed by "Failed" instead of "Pass". Please note down all messages upon a failure so that the factory or an application engineer can assist you in debugging the system. Each test button is explained in more detail below :

Test All

Tests all the blocks sequentially.

Read System Info

A detailed configuration information of your ICEPIC system is displayed as shown below :



Name	"ICEPIC" is displayed.
Series	"A", "B", etc, depending on your system
Hardware Version	Displays the hardware (board) revision number
Firmware Version	Displays the controller Firmware Version number
Trace Support	Displays YES or NO depending on whether your system supports trace.
Target Board	Displays all the processors your daughter board supports.
Target ID	An ID number (HEX) is displayed. This information is for factory testing only.
ID String	An ID string of your system. This information is for factory testing only.

Test Logic

Tests various functions of the system logic. If the test fails, please note down any errors reported and call the factory.

Test Emulation Memory

Tests the emulation RAMs in your ICEPIC system and reports if any errors found. Every location of the memory is tested and if an error is encountered on any location, it's address, expected data and the actual data are reported.

Test Break Points

The break point logic is tested. Both the system logic related functions and the emulator bond out chips are tested.

Test Trace Logic

This button is enabled only if your system has hardware trace support.

Test Target Processor

The target processor is tested for various functions. The user must make sure the system is not connected to his/her target application, since various I/O pins & Special Function Registers are modified.

Command Line Parameters

The installation program sets the command line parameters for the diagnostics program during installation. However, if at a later time, if the ICEPIC is connected to a different Comm Port, the following parameters must set :

-C# where # is the Comm Port number (1, 2, 3, etc.)

-B# where # is the desired baud rate (2400, 4800, 9600, 19200, 38400 or 57600)

Example : If your ICEPIC system is connected to COM2 and you wish to test it at 9600 baud, then the correct command line is

icediags.exe -c2 -b9600

Appendix A : Text Editor & Source Level Debug

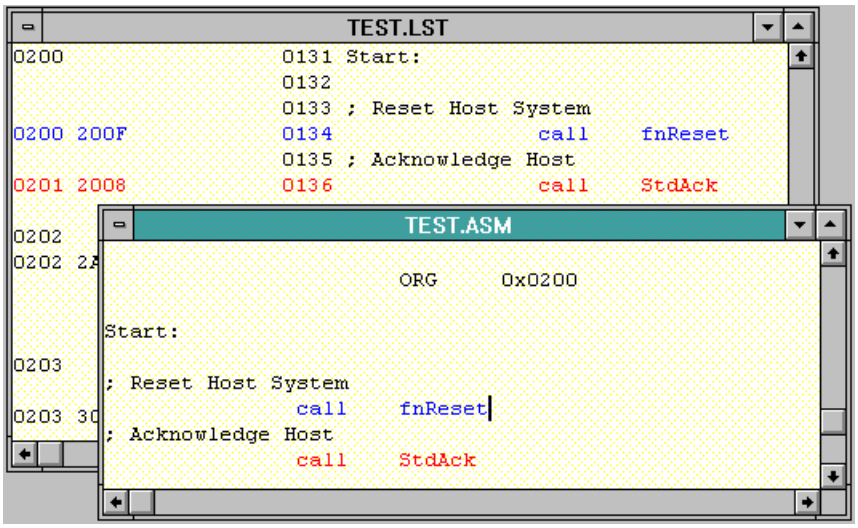
Your ICEPIC system has a full featured built-in text editor. The editor is similar to the standard Windows Notepad program in terms of usage but is a lot more sophisticated. Unlike the notepad program which has a limit of about 32K file size, this editor is limited only by the total available memory (Windows memory, including the virtual memory). Also unlike the notepad, this editor can load multiple documents.

The editor is used not only as a program editor, but also as the source level debugger. Once a COD file is loaded, all the editor files that belong to the project have capabilities of both editing and source level debugging. In addition to normal editing, the source level debugging has the following features :

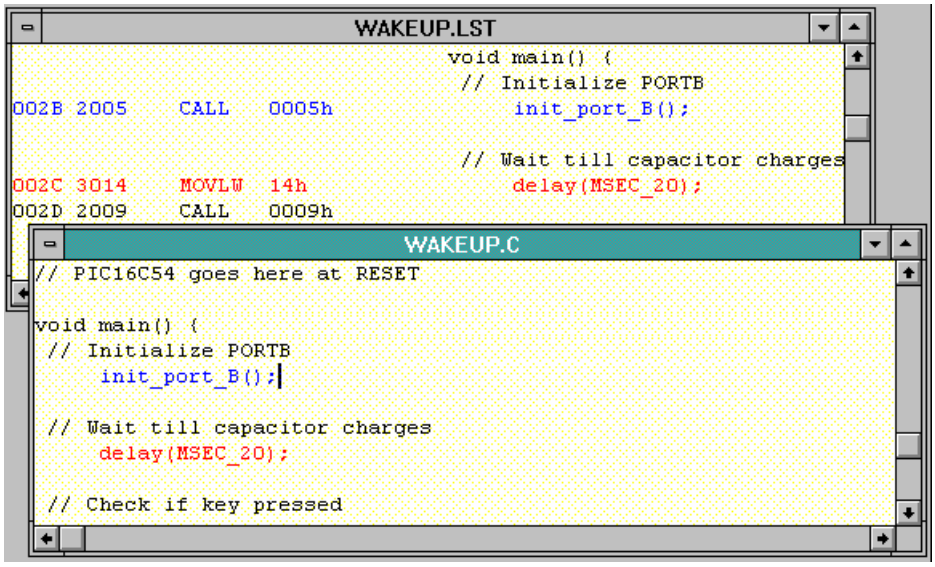
Source Debug Features

- Color coding (BLUE) of source line indicating the Current Program Counter
- Color coding (RED) of source line indicating if a break point is set
- Highlight the entire line when the Message Window is used to locate Errors/Warnings/Messages
- Listing file showing both Assembly & source/macros/C-Code
- Ability to set/clear break points both in the source files as well as the listing file
- Assemble/Compile the current source file.
- Single step and Run from the source and listing files
- Add watch points directly from the source code with a single key stroke
- Complex Text Searching and replacing while debugging

A typical source file debug of an Assembly Language (MPASM/MPASMWIN) Program along with the listing file debug is shown below :



A typical source file debug of C-Program (MPC) Program along with the listing file debug is shown below :


WAKEUP.LST
void main() {
// Initialize PORTB
002B 2005 CALL 0005h
init_port_B();
// Wait till capacitor charges
002C 3014 MOVLW 14h
delay(MSEC_20);
002D 2009 CALL 0009h
WAKEUP.C
// PIC16C54 goes here at RESET
void main() {
// Initialize PORTB
init_port_B();
// Wait till capacitor charges
delay(MSEC_20);
// Check if key pressed

Text Searching

The internal editor has complex text searching capabilities both while editing and debugging. The editor allows you to search forwards or backwards through a file for a certain "pattern" of text. In addition to searching for a series of characters, you can search for text which has a certain "property". For instance, you can ask the editor to locate all words which begin with the letter ‘Q’, or for all occurrences of three digits in a row. The search command uses a very powerful mechanism called "regular expressions" to accomplish this.

The use of regular expressions has become very popular since the use of the UNIX operating system has become widespread. Although most operating systems (such as DOS) let you specify wild cards in the command line, UNIX carries the class of expressions further. For a good discussion of regular expressions, please see the book "Software Tools" by Kerningham and Ritchie, published by Addison Wesley. (UNIX is a trademark of AT&T.)

Metacharacters

When you invoke the search command, the editor will ask you for the text to search for. Certain characters have a special meaning in the search pattern that you type. These characters are referred to as "metacharacters". Metacharacters should not be an alien concept to you. You have used metacharacters in DOS when you use the command "DIR R*.DOC"; the asterisk is a metacharacter which means "match 0 or more characters".

Here is a list of metacharacters which you can use when you type a pattern to search for :

<c> Matches the character <c>. For example, abc will match an ‘a’ followed by a ‘b’ followed by a ‘c’.

? Matches any single character . For example, a?c will match an ‘a’ followed by any character followed by a ‘c’.

* Matches zero or more occurrences of the previous character or metacharacter. For instance, ab* will find an ‘a’ followed by zero or more b’s.

+ Matches one or more occurrences of the previous character or metacharacter For instance, ab+ will find an ‘a’ followed by one or more b’s.

[c1-c2] A character class. Matches any character within the class. You can list any characters within the class. If there is a dash encountered in the class, then all characters starting from the character before the dash and ending with the character after the dash will be included in the class. For instance, [a-e] will match ‘a’,’b’,’c’,’d’, or ‘e’. [a-ex-z] will match ‘a’ through ‘e’ or ‘x’ through ‘z’.

[^c1-c2] Matches any character NOT IN the class. For instance, [^a-z] will match any character which is not in lower case.

\ Takes the next character literally. This is called an "escape" character. If the next character after the backslash is a ‘c’, a digit, or a parenthesis, the sequence is equivalent to one of the metacharacters discussed below.

^ Anchors the match at the beginning of the line. For instance, ^abc finds a line which begins with "abc".

$ The match succeeds if at the end of the line. For instance, abc$ finds a line which ends with "abc".

a|b Matches pattern a or pattern b.

\c If the match was successful, places the cursor at that position.

\(a\) Tags pattern <a>. This tag can be referred to in subsequent parts of the search pattern or in the substitution pattern.

\n Refers to the tagged pattern <n>. N is a number from 1 to 9.

{ } Parenthesizes an expression

The following shortcut patterns have been pre defined and are available for you to use:

	Meaning	Equivalent pattern
:a	alphanumeric character	[a-zA-Z0-9]
:b	whitespace character	[\t]+
:c	alphabetic character	[a-zA-Z]
:d	numeric character	[0-9]
:h	hex number	[0-9a-fA-F]+
:i	integer	[0-9]+
:l	valid C identifier	[a-zA-Z_][a-zA-Z_0-9]+
:w	word	[a-zA-Z]+

IMPORTANT - To search for a colon, you must prefix it with a backslash (i.e. - \:).

EXAMPLES OF SEARCH PATTERNS

abc	Matches ‘abc’.
^$	Matches a line with no characters. (the beginning of the line followed by the end of the line)
^abc	Matches a line which starts with ‘abc’.
^abc$	Matches a line which contains only ‘abc’.
[A-Z][0-9]	Matches a capital letter followed by a digit.
[^A-Za-z0-9]	Matches a character which is not alphanumeric.
be?t	Matches ‘be’, followed by any character, followed by a ’t’. (i.e. - bert, belt, bent).
ab*c	Matches an ‘a’, followed by zero or more ‘b’s, followed by a ‘c’.
ab*\cc	Same as above, but places the cursor over the ‘c’ after the match is completed. The "\c" places the cursor at the position after "ab*" is matched.
ab*c	Matches ‘ab*c’. Note that "*" tells the pattern matcher to take the asterisk literally.
{abc}+	Searches for one or more occurrences of "abc".
[A-Z][A-Z]*[.?,!]	Matches a capitalized word.
if|else	Matches the word ‘if’ or the word ‘else’.
if|else|for	Matches ‘if’ or ‘else’ or ‘for’.
\(abc\)[0-9]\1	Matches ‘abc’, any digit, then ‘abc’. Note that "\1" refers to tagged pattern 1, which is "abc".
\([A-Z]+\)?*\1	Matches a line which contains two or more occurrences of the same upper-case word. Note that "\1" refers to whatever [A-Z]+ matched.

In order to search for a character which also doubles as a metacharacter, you must prefix that character with a backslash. For instance, if you wanted to search for a question mark, the search pattern would be "\?". To search for a backslash, the pattern is "\\".

Metacharacters in the Substitution Text

In the replacement pattern, you may use the metacharacter ‘&’. Whenever an ‘&’ is encountered in the replacement pattern, the entire matched pattern will be substituted at that point.

For example, let’s say that you specified ‘abc’ as the pattern to search for. A replacement pattern of ‘&xxx’ would change ‘abc’ to ‘abcxxx’. Likewise, a replacement pattern of ’12 && 34' would change ‘abc’ to ’12 abcabc 34'. To use a ‘&’ literally in the replacement pattern, precede the ‘&’ by a backslash (’\’).

You may also include a tagged pattern in the replacement pattern. For example, let’s say that the pattern that you searched for was :

```
\(dog\) = \(cat\)
```

If you want to replace every occurrence of "dog = cat" with "cat = dog", then the replacement pattern that you specify is :

```
\2 = \1
```

This says that you want the new replacement pattern to be the first matched pattern ("dog") followed by " = ", followed by the second matched pattern ("cat").

To show a more powerful example, you can have the search pattern :

```
\([a-z]+\)->prev = \([a-z]+\);
```

and the substitution pattern :

```
\2->next = \1;
```

The pattern will match a sequence of lower case letters, followed by "->next = ", followed by another sequence of lower case letters, followed by a semi-colon. The first sequence of lower case letters that is matched is tagged as "\1", and the second sequence of lower case letters is tagged as "\2". This would change the line

```
currentline->prev = newline;
```

to

```
newline->next = currentline;
```

Editor Keyboard Commands

HOME	Move cursor to start of line
CTRL+HOME	Move cursor to first character in buffer
END	Move cursor to end of line
CTRL+END	Move cursor to last character in buffer
PGUP	Scroll up one page
CTRL+PGUP	Move cursor to previous paragraph
PGDN	Scroll down one page
CTRL+PGDN	Move cursor to next paragraph
LEFT	Move left one character
CTRL+LEFT	Move to the previous word
RIGHT	Move right one character
CTRL+RIGHT	Move to the next word
UP	Move cursor up one line
CTRL+UP	Move to first character in window
DOWN	Move cursor down one line
CTRL+DOWN	Move to the bottom left corner
INSERT	Toggle insert/overstrike mode
SH+INSERT	Paste contents of clipboard
CTRL+INSERT	Copy marked text
DELETE	Delete the current character
SH+DELETE	Delete the selection or current line
CTRL+DELETE	Deletes the current word
TAB	Indent (if lines marked)
SH+TAB	Unindent (if lines marked)
BACKSPACE	Backspace over previous character
CTRL+'V'	Paste contents of clipboard
CTRL+'X'	Cut into the clipboard
CTRL+'Z'	Undo last editing operation
F3	Repeat last search

Appendix B : Programming ICEPIC

Introduction

The ICEPIC software can be an "automation server" program. An "automation server" is an application that exposes programmable objects to other applications, which are called ""automation clients" (e.g.: Excel & MS Word). Exposing programmable objects enables clients to "automate" certain functions by directly accessing those objects and using the services they make available.

The ICEPIC exposes certain functions like "Run", "Step", "Halt" etc. as ICEPIC objects. For users who want to program ICEPIC to perform certain customized jobs, these exposed objects may be very useful. For example, one may wish to continuously single step the ICEPIC system until a certain RAM location is equal to the desired value with additional condition that the Program Counter (PC) is equal to a certain value. Using these exposed objects one can program to perform complex sequences of tasks. This application note describes all the exposed objects and their syntax. Example(s) are provided to control ICEPIC using Excel 5.0 macros (using Excels built in Basic language).

Exposed Objects

Listed below are the exposed objects :

- "ProcessorCmd"
- "GetPC"
- "GetRam"
- "SetRam"
- "IsHalted"
- "DownLoad"
- "FillCodeMem"
- "BreakPoint"
- "ClearBreaks"
- "GetEEprom"
- "SetEEprom"

- EXCEL 5 Example
- Automatic Data Links

References

- EXCEL 5 VBA Users Guide
- EXCEL 5 VBA On Line Help
- Inside OLE 2, Kraig Brockschmidt, Microsoft Press

ProcessorCmd

Purpose

To control the ICEPIC's execution : Run, single step, reset, etc. as shown below.

Parameters

A 2 byte command with the following meaning :

Command	Operation
0	Run the processor in real time
1	Halt the processor
2	Reset the processor
3	Single Step the processor
4	Single Step Over current line
5	Single Step Out Of current routine
6	Run ICEPIC in animation mode

Return Value

A 2 byte value

1 Successful

0 Fail

Excel Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
ICEPIC.ProcessorCmd (0)          // Runs the processor in real time
ICEPIC.ProcessorCmd (1)          // Halt the processor
```

GetPC

Purpose

To get the current value of the program counter.

Parameters

None

Return Value

A 2 byte value = current program counter value

-1 indicates error

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
pc = ICEPIC.GetPC
```

GetRam

Purpose

To get the Data Ram value at desired RAM address.

Parameters

1. Address of the desired RAM Location (a 2 byte value)

Return Value

A 2 byte value = Data of the requested RAM address

-1 indicates error

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
Data = ICEPIC.GetRam(Addr)
```

SetRam

Purpose

To modify the data of a certain Data RAM (File registers) location

Parameters

1. RAM address (a 2 byte value)
2. RAM Data (a 2 byte value)

Return Value

1 = Successful

0 = Error

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
bSuccess = ICEPIC.SetRam(addr, data)
```

IsHalted

Purpose

To check if the processor is halted or is still running. This status check is useful when writing some custom macros. For example, one may program to Run the processor and then check to see if the processor is halted before a Data Ram location is read.

Parameters

None

Return Value

1 = Halted

0 = Not Halted

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
bHalted = ICEPIC.IsHalted
```

DownLoad

Purpose

To download a COD File to the emulator. The action is same as using the File Download command from the ICEPIC.

Parameters

1. A string containing the name of the file to be downloaded. The path and file name must be valid. The file extension is optional, defaulting to COD.

Return Value

1 = Successful

0 = Error

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
bSuccess = ICEPIC.DownLoad("c:\oletest\fun.cod")
```

FillCodeMem

Purpose

To modify the program memory location(s) with the desired opcode.

Parameters

1. A 2 byte value for the Start address of the program memory to be modified
2. A 2 byte value for the End address of the program memory to be modified
3. A 2 byte value of the new desired opcode to be filled with

Return Value

1 = Successful

0 = Error

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
bSuccess = ICEPIC.FillCodeMem(10, 20, 0)
```

This instruction will fill code memory from address 10 to 20 with zeros (NOPs)

BreakPoint

Purpose

To set or clear a program memory address break point

Parameters

1. A 2 byte value of the program memory address at which the break point is to be set/cleared
2. Pass on 1 to set a breakpoint or a 0 to clear.

Return Value

1 = Successful

0 = Error

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
bSuccess = ICEPIC.BreakPoint(100, 1)
bSuccess = ICEPIC.BreakPoint(101, 0)
```

This instruction sets a breakpoint at address 100 and clears a breakpoint at address 101

ClearBreaks

Purpose

Removes all breakpoints

Parameters

None

Return Value

None

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
ICEPIC.ClearBreaks
```

GetEEprom

Purpose

To get the Data Ram value of a certain address.

Parameters

1. EEPROM address (a 2 byte value)

Return Value

A 2 byte value = EEPROM Data at the requested address

-1 indicates error

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
EEdata = ICEPIC.GetEEprom(EEaddr)
```

SetEEprom

Purpose

To modify the data of a certain EEPROM Data Memory Location. Note that all PIC16CXX processors do not have a EEPROM Data memory. Currently only PIC16C84 has this memory

Parameters

1. EEPROM address (a 2 byte value)
2. EEPROM Data (2 byte value)

Return Value

1 = Successful

0 = Error

Example

```
Dim ICEPIC As Object
Set ICEPIC = CreateObject("ICEPIC.Automation")
bSuccess = ICEPIC.SetEEprom(EEaddr, EEdata)
```

Excel Automation Example

A complete example to program ICEPIC from Excel 5.0 is shown below. This example shows Buttons attached to various excel macros. Using this example, you can create more complex operations as your needs require. This example file is located under "excel" sub-directory in "ICEPIC" directory.

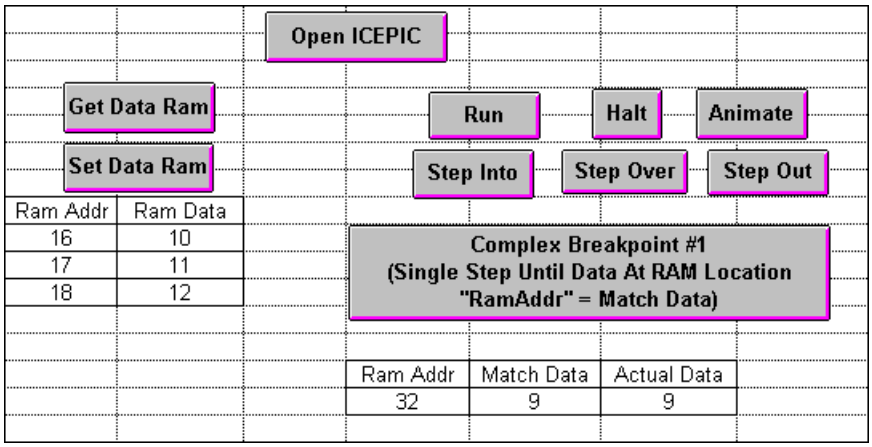


Fig 1.1 An Example Excel Spreadsheet

Visual Basic Code (VBA)

```
Dim ICEPIC As Object

Sub Form_Load()
   Set ICEPIC = CreateObject("ICEPIC.Automation")
End Sub

Sub Close_Click()
   Set ICEPIC = Nothing
End Sub

Sub Run_Click()
   ICEPIC.ProcessorCmd (0)
End Sub

Sub Halt_Click()
   ICEPIC.ProcessorCmd (1)
End Sub

Sub Reset_Click()
   ICEPIC.ProcessorCmd (2)
```

```
End Sub

Sub StepInto_Click()
   ICEPIC.ProcessorCmd (3)
End Sub

Sub StepOver_Click()
   ICEPIC.ProcessorCmd (4)
End Sub

Sub StepOut_Click()
   ICEPIC.ProcessorCmd (5)
End Sub

Sub Animate_Click()
   ICEPIC.ProcessorCmd (6)
End Sub

Sub ClearBreaks_Click()
   ICEPIC.ClearBreaks
End Sub

Sub DownLoad_Click()
   ICEPIC.DownLoad (CodFileName.Text)
End Sub

Sub GetPC_Click()
   ActiveCell.Value = ICEPIC.GetPC()
End Sub

Sub GetRamData_Click()
   Do Until ActiveCell.Value = ""
      RamData = ICEPIC.GetRam(ActiveCell.Value)
      Selection.Offset(0, 1).Value = RamData
      Selection.Offset(1, 0).Range("A1").Select
   Loop
End Sub

Sub SetRam_Click()
   Do Until ActiveCell.Value = ""
```

```
        RamError = ICEPIC.SetRam(ActiveCell.Value, Selection.Offset(0, 1).Value)
        Selection.Offset(1, 0).Range("A1").Select
        If RamError = 0 Then
           Beep
        End If
    Loop
End Sub

Sub ComplexBreak_1_Click()
    ActualData = ICEPIC.GetRam(ActiveCell.Value)
    MatchData = Selection.Offset(0, 1).Value
    Do Until ActualData = MatchData
        ICEPIC.ProcessorCmd (3)
        ActualData = ICEPIC.GetRam(ActiveCell.Value)
        Selection.Offset(0, 2).Value = ActualData
    Loop
End Sub
```

Fig 1.2 The Visual Basic Code (VBA macros) For Spreadsheet

If you want to create the above example by yourself, follow the steps shown below:

1. Start Excel 5.0 or greater with a new workbook, choose Macro from the Insert menu, and choose Module.
2. Type in the macro code shown in Fig 1.2
3. Click the Sheet1 tab, choose Toolbars from the View menu. Check the Forms Check Box to display the Forms toolbar. Click OK.
4. Click the Create Button control and drag in the upper left corner (or anywhere you please) of the worksheet to create a push-button. Assign the button to the "Form_Load" macro
5. Size the push-button and type the caption "Open ICEPIC" as shown in Fig 1.1
6. Similarly add pushbuttons and assign them to the appropriate macros as shown in Fig1.1
7. Type some numbers under "Ram Addr" (in Column 1) and press the "Get Data Ram" button. The value of data ram will be now be updated.
8. Similarly try pressing the single step button and note that ICEPIC executes a single step.
9. Try the "Complex Breakpoint #1" and modify it to perform various complex tasks which can be modified to suit your specific needs.

Automatic Data Links

In addition to above mentioned OLE Automation features, ICEPIC also supports OLE Automatic Links to all the Data File Registers. Links are supported through standard

Windows Clipboard. To copy the data file registers to clipboard, select the Data File Registers window and perform Copy (or CTRL + C) from the Edit menu. The data can now be pasted into any other Windows application. For those applications supporting Links (like EXCEL 5 or Word 6 or Lotus for Windows), use Paste Special or Paste Link command to establish hot (or active) links to ICEPIC's Data File Registers. By establishing links, any change in Data Ram in ICEPIC will automatically be updated in the pasted application.

An example of this link using EXCEL5 is shown below in Fig 2.1

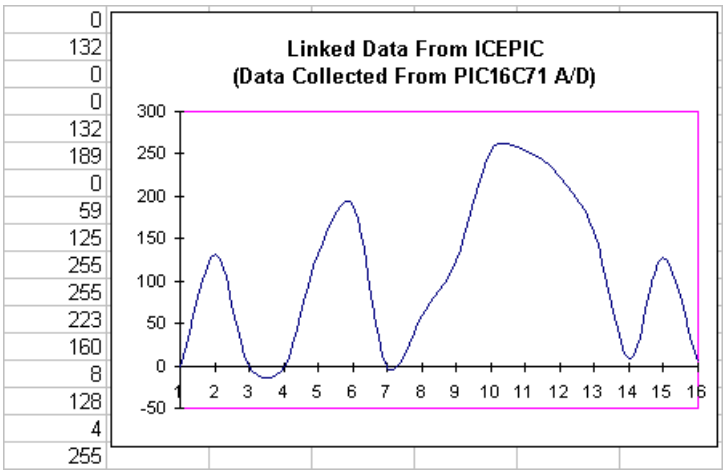


Fig 2.1 An Example Showing Hot Linked Data In EXCEL Spreadsheet

NOTE : Selecting "Paste" pastes only the data without any links. To link the data to ICEPIC, from EXCEL, select the Paste Special... menu (under Edit) and select the Link option.

Single step the ICEPIC and note that any change in the Data File Registers in ICEPIC is automatically reflected in EXCEL (both in the Worksheet as well as the Graph).

Appendix C: Additional Features of ICEPIC2

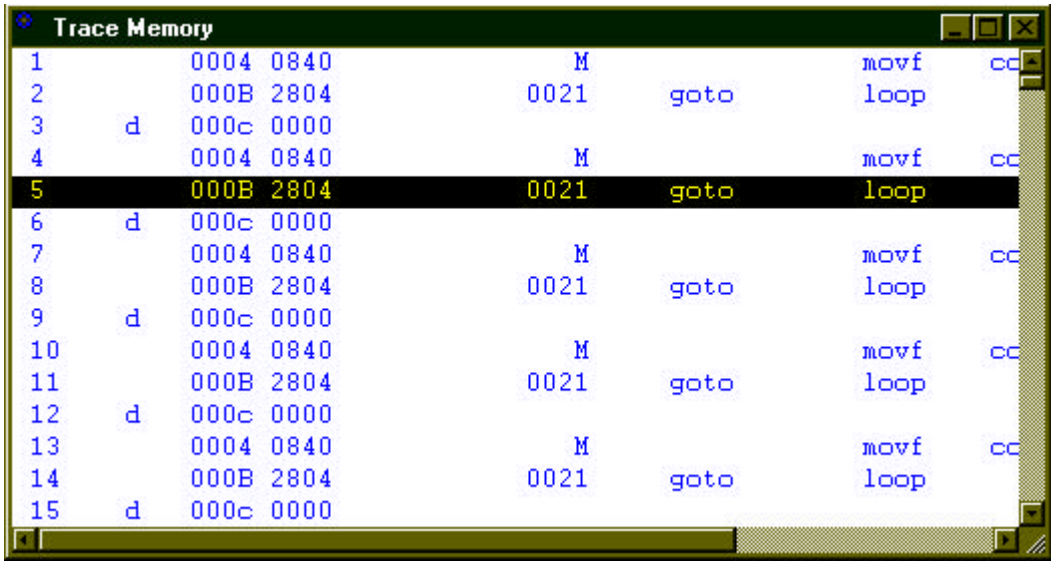
In addition to all the features found in ICEPIC, the ICEPIC-II system has (a) Real Time Trace and (b) Trigger Output. The real time trace buffer is 4K deep (circular) and logs on the program flow in Real Time. This document describes the features of ICEPIC-II and is an add-on document to the existing ICEPIC User's manual. Only the features not found in ICEPIC are described here.

Real Time Trace

The ICEPIC-II has a trace buffer of 4K. The buffer is circular, meaning that at any time the buffer is overflowed, the past data is overwritten. If the buffer is overwritten, the user can view the latest 4K of data.

Any number of Hardware trace points can be set within the program memory space. The user can select the desired section of program memory to be traced.

A sample view of the trace buffer window (opened by "View Real Time Trace" menu) is shown below :



The 1st column displays the buffer address, the 2nd column displays "d" if the cycle is a dummy fetch (on a 2 cycle instruction, like GOTO or CALL). The data displayed will be either

Source code ; If the LIST file is open, or

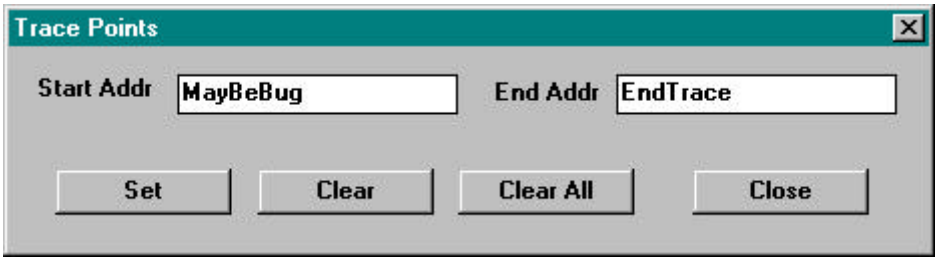
Dissassembled object code ; if the LIST file is not opened

It recommended to load the LIST file (see Options, Code Download ..., Open List File menu).

NOTE : The dummy cycle (or 2nd cycle) as shown above can be eliminated from the hardware trace buffer, by unchecking the "Capture 2nd Cycle" in the "Options Trace" menu. Capturing the 2nd cycle is useful if timing (cycles) are to be measured.

Setting/Clearing Trace Points

1. To set a Trace point at the desired address, goto the Code Window and hi-light the desired address. Press Ctrl + T or goto “Qualifiers” menu and select “Toggle Trace Point”
1. To set/clear trace points over an address range, goto “Qualifiers” menu & select “Trace Range ..”. A dialog box as shown below appears :



2. Type in the desired address range and select “Set” to set trace points in the selected address range. Clear button clears the tracepoints in the desired address range. Clear All button clears all trace points, irrespective of the entered address range.

In most applications the 4K buffer is adequate. However the trace buffer may be increased to 8K by a factory upgrade. Please contact RF Solutions if this is required.

Trigger Output

The ICEPIC-II has a Trigger Output located on the motherboard between U8 & U9 (at the rear left hand side).

The trigger output provides a digital output which can be connected a scope or any other device to timing measurements, etc.

The trigger out may be configured to trigger on any address or address ranges, in the same way as a break point is set.

The Trigger points can be set using (1) Ctrl + O from the Code Memory Window (b) by selecting the “Qualifiers, Trigger Out Range..” menu.

Appendix D: Troubleshooting

Whenever an error situation occurs, please check power, connections, the selected Comm Port and perform a System Wide Reset (in "Debug - System Reset" menu). The most commonly encountered errors are described below:

Serial Communication Time-out

This error message is reported on various circumstances. To circumvent this

- Make sure the correct Comm port is selected. A free Comm port may have been selected, but ICEPIC is connected to another serial port.
- The baud rate may be too high for your system. See Appendix B on how to achieve high baud rates.

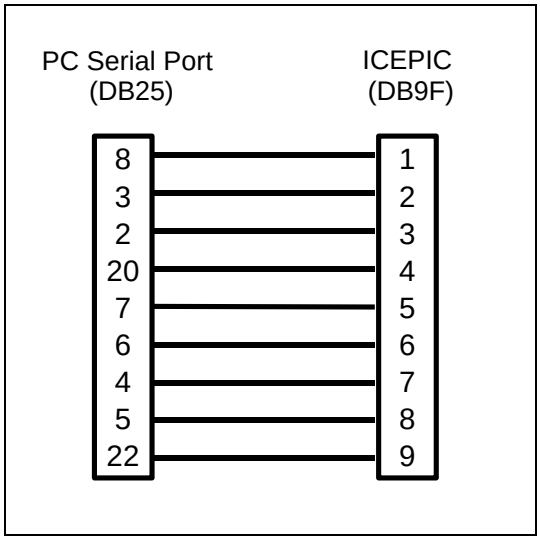
Processor Not Responding

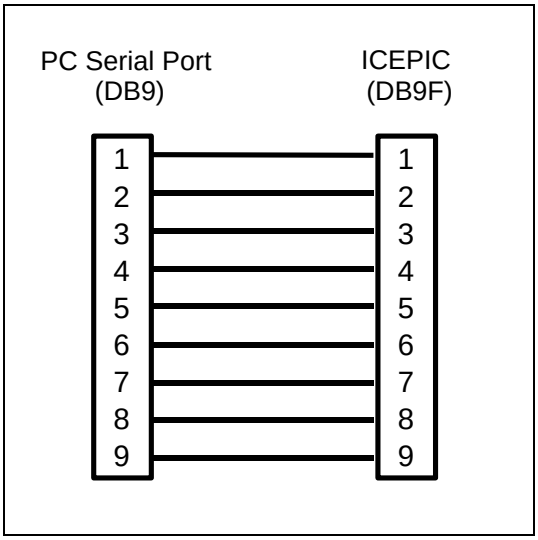
This error message is reported whenever the target processor does not respond to various commands. This may occur on the following situations :

- The MCLR pin is held low. If the ICEPIC system is plugged into the target application, make sure the MCLR pin is not held low. If the ICEPIC system is not connected to the target application, make sure the MCLR pull-up jumper (Link 3, LK3) on the daughter board is enabled.
- Clock may be absent. Make sure the Clock Oscillator on the daughter board is connected. ICEPIC system cannot read or control the target processor without this clock.
- Attempt to halt while the processor is in sleep mode. If the processor is in sleep mode (by executing a SLEEP instruction), the only way to make the processor halt is to bring the processor out of sleep mode (reset, interrupt, etc.).

Appendix E: Serial Communication

The ICEPIC interfaces to the host computer via RS232 connection. A standard DB9 connector is used. The following tables give the connection requirements;





Comm Port	Any port supported by MS Windows. The port is selectable from the software. No dip settings are necessary.
Baud Rate	2400 bps to 57600 bps
	Selected from software. No dip switches need to be set. The emulator system detects baud rate automatically
Parity	None. Fixed, cannot be selected by the user
Data Bits	Fixed to 8 bits. Cannot be set by the user
Stop Bits	1 Stop Bit. Cannot be selected by the user
Software Receive Buffer	Fixed to 2048 bytes
Software Transmit Buffer	Fixed to 2048 bytes
Firmware Buffer	Fixed to 64 bytes

How to Achieve High Baud Rates

- **Use an I/O Card with 16550 processor.**
  Older IBM PC compatibles were equipped standard with 8250 UART. Newer computers come with 16550 processors (esp. notebook computers). Unlike 8250, the 16550 UARTs have on board 16 byte FIFO which improves the speed of communications. MS Windows will have enough time before servicing an interrupt.
- **Upgrade your Windows 3.1 to Windows For Workgroups**
  Even if your computer is not networked, Windows For Workgroups' serial port driver is significantly better.
- **Upgrade you computer to at least 8 Mb RAM.**
- **Remove any screen savers, which slows down your Windows operation.**

There are various third party COMM Drivers on the market, which are replacements for the standard driver that comes with MS Windows. These drivers are supposedly faster, but ICEPIC software has not been tested with these drivers.

What's Your UART ?

- At your DOS prompt (e.g. C:\), enter MSD.EXE
- Choose Comm Ports
- If your serial I/O card is equipped with old technology, the screen will show 8250 or 16C450
- If you have a high speed UART, the screen will show 16C550

Appendix F: Data Types and Formats

ICEPIC supports the following data types, that can be used on data file registers to watch and modify.

BYTE	8 Bit unsigned value. Represents 1 byte
	Range 0 to 255
CHAR	8 Bit value represented as a char
INT	16 Bit signed value. Represents 2 bytes
	Range: -32768 to 32767
UINT	16 Bit unsigned value. Represents 2 bytes
	Range: 0 to 65535
LONG	32 Bit signed value. Represents 4 bytes
	Range: - 2147483648 to 2147483647
ULONG	32 Bit unsigned value. Represents 4 bytes
	Range: 0 to 4294967295

The multibyte data can be represented in either Big Endian or Little Endian formats. These formats are explained below:

Big Endian

High bytes are located at higher address space, low bytes are at lower address. For example, if "X_16Bit" symbol is represented as UINT, and if X_16Bit equals 0x55AA, and if it's address is 0x10, then X_16Bit is organized as follows :

LOBYTE(X_16Bit)	=	0xAA	located at address 0x10
HIBYTE(X_16Bit)	=	0x55	located at address 0x11

Little Endian

High bytes are located at lower address. Taking the same example as in described above, the data is organized as follows:

LOBYTE(X_16Bit)	=	0xAA	located at address 0x10
HIBYTE(X_16Bit)	=	0x55	located at address 0x0F

Appendix G : Power Supply

The ICEPIC system is shipped with its own power supply. If the user wishes to connect his/her own power, make sure the following specifications are met :

Input Voltage	:	9 - 12 Volts (AC or DC)
Current	:	500 mA Minimum.

NOTE: The ICEPIC system's logic does not support 3V emulation. The system strictly requires 5V ± 5%

The target probe cable which connects between the personality daughterboard and the target board does contains the connections of the PIC I/O lines and the 0Volt (GND) to the target PIC. Please note that the Vcc connection is not brought out to the target probe header.

Warning!

Care must be taken to avoid exposing the Target probe cable to any excessive static or reverse voltage.

The Microchip emulation devices (16C01/2, 16CXXME etc) do not contain the same high voltage protection on their I/O lines as the standard PIC Microcontrollers.

Extra care is therefore advised when handling the ICEPIC system, in particular the

Appendix H : User Configuration Links

For Daughter boards not covered in this manual please read “Links.txt” under your c:\icepic\ directory for the latest information.

OSC Mod Config

These links configure the system emulator chip and for normal operation with the OSC Module supplied they should be left open.

For all PIC’s pins ‘OSC1’ & ‘OSC2’ are used to connect to the oscillator source. ‘OSC1’ is always an input but ‘OSC2’ can be input or output. The only mode where ‘OSC2’ is an output is RC Mode. Then, the output from ‘OSC2’ is 1/4 of the frequency at ‘OSC1’

The function of the 16C01 or 16C02 is the same as for any PIC with the mode of operation being set by the two Links SOSC1, & SOSC2, on the personality daughter board, just above the OSC Module.

SOSC1	SOSC2	Mode
Open	Open	**RC**
Open	Closed	**HS**
Closed	Open	**XT**
Closed	Closed	**LP**

The ICEPIC Daughter board is driven from an ‘EXT’ source (CMOS/TTL output Oscillator Module) which feeds into ‘OSC1’ so if you connect the Links to RC mode you should see the 1/4 frequency on ‘OSC2’.

Caution must be observed if the target probe is connected the target board and you have additional oscillator circuitry on the target board!

MCLR

This link connects a weak pull-up resistor (27kΩ) to the emulator MCLR input.

This link must be connected, to enable the emulator to operate, unless a target application circuit is connected with a working pull up resistor, or reset circuit. It may be removed when running in real time, thus truly emulating a reset condition.

Generally, the MCLR link may be left connected for normal operation with or without a target application board connected.

OSC Input - Internal or External Source

This link enables the user to select the clock from either the on-board User changeable OSC Module (‘INT’ position), or from the clock on OSC1 pin 1 of the users target board (‘EXT’ position).

Due to the inherent nature of emulators and cables, use of an XTAL on a users target board cannot be guaranteed. For reliable operation we recommend using the ‘INT’ option.

ICEPIC & ICEPIC2 Daughter Board Link Configuration

Daughter Board	SOSC1 & SOSC2	MCLR	OSC I/P	Link 6	See Note
ICEPIC-DB5X	Link 1 & 2	Link 3	N/A	N/A	
ICEPIC-DB55X	Link 1 & 2	Link 3	N/A	Must be connected between ‘2’ & ‘3’	
ICEPIC-DB62X	Link 1 & 2	Link 3	Link 7	Must be connected between ‘2’ & ‘3’	
ICEPIC-DB711	Link 1 & 2	Link 3	Link 7	Must be connected between ‘2’ & ‘3’ For emulation of 16C61 (to turn off the A/D) set Link between ‘1’ & ‘2’	
ICEPIC-DB715	Link 1 & 2	Link 3	Link 7	Must be connected between ‘2’ & ‘3’	1
ICEPIC-DB74A	Link 1 & 2	Link 3	Link 7	N/A	1
ICEPIC-DB77	Link 1 & 2	Link 3	Link 7	N/A	1
ICEPIC-DB84A	Link 1 & 2	Link 3	Link 7	Always be connected between ‘1’ & ‘2’	
ICEPIC-DB92X	Link 4 & 5	Link 3	J1	N/A	2 & 3
ICEPIC2-DB17	J 2 & 3	J5	J1	N/A	

Notes

1. When using the A/D converter, the system +5V supply is connected to the Analogue Vdd (Avdd). If the user wishes to supply a higher precision, low noise voltage to the AVdd pin, then the link may be removed, and the external Analogue Power Supply fed in directly to pin 2. Care must be taken not to exceed the normal power ratings of the AVdd pin.
2. Jumper Link J7 is currently unused
3. J8 provides a link to enable the “OSC2/CLKOUT” (centre pin ‘2’) signal of the target probe cable to be connected to either the “OSC2/CLKOUT” (pin ‘1’) of the 16C02, or the OSC module output (pin ‘1’). This provides a direct output of the OSC Module signal to the target cable.

ICEPIC-Jnr-5X Link Configuration

OSC Mod Link A & B	MCLR	OSC I/P	Link 6
Link 1 & 2	Link 3	N/A	N/A

Emulation of PIC12C508 & 509

The user must note that there is no special bond out chip available for the 12C50x processors. Because of this, it is impossible to emulate certain functions. A list of these functions NOT emulatable by ICEPIC is given below.

Four of the GPIO pins (general purpose I/O pins) are multiplexed with other functions. The functions of these four pins is selected by FUSE options in the 12C50x processors. However it is not possible to emulate these fuses in an emulator. To emulate the muxing of these pins, jumper links are provided to select the option desired by the user.

Link Settings :

Link	PIC12C Pin No	Function when link Set between '1' & '2'	Function when link Set between '2' & '3'
S1	2	RB5	OSCIN
S2	3	RB4	OSCOUT
S3	4	RB3	MCLR
S4	5	RB2	RTCC

S5	4	Connects Pull-up resistor on pin 4
S6	6	Connects Pull-up resistor on pin 6
S7	7	Connects Pull-up resistor on pin 7

Non emulatable functions :

1. Software selection (through OPTION register) of weak pull-ups of GP0 & GP1.
2. The wake-up on change feature on GP0 & GP1
3. GP3 on 12C50x is input only pin. In the emulator it is an I/O pin. Care must be taken by the user not to configure this pin as OUTPUT.
4. The internal RS Oscillator is not emulatable. As such, the calibration , etc related to this internal clock are not emulatable. The user must use the 4 Mhz canned clock supplied by the 5x daughter board.
5. MCLR Filtering improvements found in 12C50x are not emulatable.

ICEPIC-Jnr-16 EmulatorBoard Link Configuration

SOSC1 & SOSC2	MCLR	OSC I/P
Link 1 & 2	Link 3	Link 7

Appendix I : Emulation Limitations

Warning!

Care must be taken to avoid exposing the Target probe cable to any excessive static or reverse voltage.

The Microchip emulation devices (16C01/2, 16CXXME etc) do not contain the same high voltage protection on their I/O lines as the standard PIC Microcontrollers.

Extra care is therefore advised when handling the ICEPIC system, in particular the ICEPIC Daughterboards.

Due to the nature of the emulator system and the emulator bond out chips, there are a few situations where true emulation is not possible. The following section describes them:

Processor Specific Limitations

- Watchdog timer is emulated only when running in real time. It is held in reset during single steps.
- Power On Reset (POR) conditions are not emulated. The bond out emulator chip is always powered on by the ICEPIC system. Thus PCON register of PIC16Cxx is not emulated. The "TO" & "PD" bits may not correctly show the power-up conditions.
- Power-up Timer is not emulated
- Oscillator Startup Timer is not emulated.
- Cannot emulate on chip EPROM programming.
- Modifying PORTB of PIC16Cxx is not immediately reflected onto the I/O pins during a HALT state if the "Freeze Peripherals On Halt" is selected. At least one single step or run in real time is necessary.
- Interrupts are not recognized during HALT state.
- When the processor is in "SLEEP" state, HALT is not recognized by the emulator system, and the user must reset the processor to halt or come out of sleep mode. As such, single step on a SLEEP instruction is not possible

ICEPIC System Specific Limitations

- The "Step Over" & "Step Out" functions may not freeze the peripherals (if selected) for 1 cycle.
- Cannot emulate in real time below 32 kHz.
- The ICEPIC is able to emulate up to a maximum of 20MHz. Please note that at the time of writing this manual ,the maximum speed of operation of the emulation chip PIC16C02ME, is 10MHz. This means that when emulating PIC microcontrollers which utilize the PIC16C02 as the core emulation device, the maximum emulation speed is 10MHz.
- Cannot emulate low voltage operation.
- Clock modes (XT, RC, HS & LP) cannot be selected. Clock is always fed from the user changeable Clock Oscillator located on the daughter board.

ICEPIC Oscillator/Clock Limitations

The probe header cable has a high capacitance, therefore is not always possible to transmit a clock signal from the target board to drive the bond out chip. (Please note this feature is not available on the 5X Personality daughter board)

If the user requires a specific clock signal, the following is recommended;

- Using an Oscillator module on the daughter board with the appropriate frequency.
- Connect a separate cable from the clock source directly into the OSC Module output (pin 8) on the daughter board, with low impedance to ensure enough drive power to the input of the emulation chip.

A kit of Oscillator modules is available from your supplier, please contact sales for further information.